EFFICIENT NUMERICAL METHOD FOR SOLUTION OF L^2 OPTIMAL MASS TRANSPORT PROBLEM

A Thesis
Presented to
The Academic Faculty

by

Tauseef ur Rehman

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2010

EFFICIENT NUMERICAL METHOD FOR SOLUTION OF L^2 OPTIMAL MASS TRANSPORT PROBLEM

Approved by:

Professor Allen Tannenbaum, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Eldad Haber, Advisor
Department of Mathematics and Computer Science
Emory University

Professor Jeff Shamma
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Anthony Yezzi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Oskar Skrinjar
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor George Biros
College of Computing
Georgia Institute of Technology

Date Approved: 03 December 2009

To my parents,

Muhammad Ashraf and Kishwar Sultana

for their unconditional love, prayers and personal sacrifices.

ACKNOWLEDGEMENTS

I would like to begin the acknowlegements by humbly thanking 'God Almighty' for giving me the courage, capabilities and oppurtunity to complete this task as well as all the other gifts in life. In addition to this, there are several people who need to be acknowledged for their help in making this thesis possible as well as support throughout my doctorate years. In particular, I would like to thank the following people:

- Prof. Allen Tannenbaum - my advisor. It is a priviledge to be counted amongst his 'Jedi apprentices'. His extraordinary intellect, knowledge and personality have inspired and changed me in numerous ways.

- Prof. Eldad Haber - my co-advisor. I want to thank him for his unfathomable patience, sharp guidance and his most valuable influence on my research. I have only admiration for his contagious optimism and pragmatic approach to life. Thank you for your friendship.

- Thesis Committee Members - I want to thank Prof. Anthony Yezzi, Prof. Jeff Shamma, Prof. Oskar Skrinjar and Prof. George Biros for their valuable inputs in my thesis review and evaluation.

- Prof. Steven Haker and Lie Zhu. For their extensive prior work on image processing applications of Optimal Mass Transport that formed the basis and inspiration for this thesis.

- Shiela Shulte and Messam Naqvi - Before I go any further I need to acknowledge these two wonderful people for the role they played in my arrival at Georgia

Tech. Without their help my journey at Tech may have not even begun. I owe a very special thanks to these two individuals.

- John, Gallagher and Jimi - My fellow Jedi apprentices at MINERVA Lab and entrepreneurially like-minded partners with whom I hope to share many more adventures in times to come.

- Other members of the MINERVA Lab (Past and Present) - Being a member of this research lab has been a very fulfilling and rewarding experience both in terms of technical collaborations and friendships. I want to thank Ponnappan Arumuganainar, Samuel Dambreville, Yi Gao, Eli Hershkovits, Peter Karsav, Ivan Kolesov, Shawn Lankton, Xavier LeFaucheur, Steven Marzec, Oleg Michailovich, Vandana Mohan, Delphine Nain, Marc Niethammer, Eric Pichon, Yogesh Rathi, Romeil Sandhu, Julien Traisnel, Patricio Vela, Amanda Wake, Yan Yang and Lei Zhu, for our multiple and fertile interactions as well as for their contribution to the joyful yet studious atmosphere in the lab.

- Friends and colleagues at Georgia Tech - I would also like to thank my many friends and colleagues at Georgia Tech. In particular, Salman Aslam for being an all-round friend for so many years. Other friends include Irteza Shah, Jehanzeb Burki, Irteza Khan, Joshua Griffin, Farooq Akram, Faisal Khan, Qaiser Chaudary, Tahir Zaidi, Adeel Khalid, Murtaza Askari, Taimoor Khawaja, Faisal Siddiqui, Azhar Hasan, Ali Hashmi, Irteza Barlas and Mudassir Nisar. I extend my deepest gratitude to all these people and others, who's names I can't recall, for their help and support over the span of my time at Georgia Tech.

- NUST, PAF and Government of Pakistan - I want to thank these organizations for providing me the oppurtunity to fulfill my academic dreams at one of the most prestigious institutes in the world. I hope and pray these programs continue to be the platform of oppurtunity for talented and hard working people

in Pakistan.

- My Family - I would like to thank both my parents for their life-long sacrifices and commitment to my future. I owe all my success to their prayers. I want to thank my brother Tanweer and sisters Umbreen and Naureen for their faith and confidence in me and taking care of our parents during my absence. I want to thank my daughters, Hamda, Samrah and Hunia for thier joyful smiles that have been the most enjoyable source of mental relaxation and stress reduction for me. They have sacrificed alot by having a 'PhD student' dad during these past few years. Last but not least, I want to thank my wife, Sumaira, for being a true companion on this journey. Without her encouragement and complete support, I would not have been able to complete this undertaking. Thank you for staying awake with me for many nights, taking care of the kids and being with me every step of the way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Optimal Mass Transport (OMT) is an important problem with numerous applications in a wide range of fields such as econometrics, fluid dynamics, automatic control, transportation, statistical physics, shape optimization, expert systems and meteorology. More recently, it has been shown to also have application in image processing such as for non-rigid registration and morphing alongside other differential and variational methods based on fluid dynamics.

In this thesis, we present a novel and efficient numerical method for the computation of the L^2 optimal mass transport mapping in two and three dimensions. Our method uses a direct variational approach. We have formulated a new projection to the constraint technique that can yield a good starting point for the method as well as a second order accurate discretization to the problem. The numerical experiments demonstrate that our algorithm yields accurate results in a relatively small number of iterations that are mesh independent.

In the first part of the thesis, we develop the theory and implementation details of our proposed method. These include the reformulation of the Monge-Kantorovich problem using a variational approach and then using a consistent discretization in conjunction with the "discretize-then-optimize" approach to solve the resulting discrete system of differential equations. We also develop advanced numerical methods such as multigrid and adaptive mesh refinement to solve the linear systems in practical time for even 3D applications. In the second part we show the methods efficacy via application to various image processing tasks. These include image registration and morphing. We present application of (OMT) to registration in the context of

medical imaging and in particular image guided therapy where registration is used to align multiple data sets with each other and with the patient. We believe that an elastic warping methodology based on the notion of mass transport is quite natural for several medical imaging applications where density can be a key measure of similarity betweend different data sets e.g. proton density based imagery provided by MR. We also present an application of two dimensional optimal mass transport algorithm to compute diffeomorphic correspondence maps between curves for geometric interpolation in an active contour based visual tracking application.

CHAPTER I

INTRODUCTION

Optimal mass transport is an important problem that has found applications in wide range of fields. Originally formulated by a French mathematician Gasper Monge in 1781, the method was given a modern formulation in work of Kantorovich. There have been a number of algorithms proposed for computing the optimal mass transport in literature. More recently, a method proposed in [47, 48] computes the optimal warp from a first order partial differential equation, which is a computational improvement over earlier proposed higher order methods and computationally complex discrete methods based on linear programming. However, at large grid sizes and especially for 3D registration the computational cost of even this method is significant. Rigorous mathematical details for their algorithm can be found in [2]. The same authors also proposed application of mass transport for image processing application in particualr for warping and registration in the context of medical imaging.

Although computationally expensive, the OMT method has a number of distinguishing characteristics that make it an attactive choice for registration and warping that include:

- No landmarks need to be specified,
- It is symmetrical (the mapping from image A to image B is the inverse of the mapping from B to A),
- Its solution is unique *i.e.* there are no local minima,
- It can register images where brightness constancy is an invalid assumption, and

- OMT is specifically designed to take into account changes in densities that result from changes in area or volume.

In this thesis we present a new computationally efficient numerical scheme for the minimizing flow approach for the computation of the optimal L_2 mass transport mapping. In contrast to the integration of a time dependent PDE proposed in [2], we employ in the present work a direct variational method.

1.1 Organization of this Thesis

This thesis is organized into the following chapters:

- *Chapter 2.* In this chapter, we provide the details of our novel and efficient numerical method for the computation of the L^2 optimal mass transport mapping in two and three dimensions. The problem is reformulated in a variational framework and a "discretize-then-optimize" methodology is used to numerically solve the resulting system of differential equations. We also employ a projection to mass preserving constraint that provides both a good starting point for the method as well as corrects for any deviations from the constraint during the optimization process. The numerical experiments demonstrate that our algorithm yields accurate results in a relatively small number of iterations that are mesh independent.

- *Chapter 3.* In this chapter, we further extend our numerical method by incorporarting multigrid approach for computing the Poisson like problem at each iteration of the AHT method which accounts for the major computational burden of the algorithm. We develop the multigrid algorithm for 3D application of OMT where computational efficiency becomes crucial for most practical applications. Moreover, we take special care to use parallelizable multigrid components in order to implement the multigrid solver and AHT update equations on Graphics Processing Unit (GPU) for near real time computations.

- *Chapter 4.* In this chapter we present a computational framework to compute the projection to mass preserving constraint using an adaptive mesh refinement approach. We also further simplify the KKT solver described in Chapter 2 by using a matrix-splitting technique. The resulting iterative scheme adds to the overall efficiency of the algorithm.

- *Chapter 5.* In this chapter, we present results of application of our method to image morphing and registration problems. We demonstrate the success of using the OMT method for computing realisitic morphs for different types of natural imagery that conventional methods struggle with. We also apply the method for medical image registration of post and intra operative MRI brain scans to detect and correct deformations resulting from sagging of the brain tissue after craniotomy and openning of the dura. We also introduce a novel method to register low resolution MRI atlases to recently acquired high resolution scans. We employ OMT as fine alignment step and register only white matter segmentations from the two data sets.

- *Chapter 6.* In this chapter, we present another application of the OMT method to visual tracking using dynamic active contours. Here OMT is used for the purpose of establishing diffeomorphic correspondence maps between estimated and measured curves that replace earlier methods based on Laplace method which involve complex domain decompositions. The OMT method, on the other hand, is an elegant solution to the correspondence problem and is applied globally on the implicit representations of the curve. We employ 2D multigrid methods computed on GPUs for fast computations for practical applications of the method.

CHAPTER II

OPTIMAL MASS TRANSPORT: THEORY AND NUMERICAL SOLUTIONS

2.1 Introduction to Optimal Mass Transport

Optimal mass transport is an important problem with applications in econometrics, fluid dynamics, automatic control, transportation, statistical physics, shape optimization, expert systems, and meteorology [79, 94]. The problem was first formulated by the civil engineer Gaspar Monge in 1781, and concerned finding the optimal way, in the sense of minimal transportation cost, of moving a pile of soil from one site to another. Much later the problem was extensively analyzed by Kantorovich [58], and so is now known as the Monge–Kantorovich problem.

There are several formulations of the problem [1, 79, 94] of varying degrees of generality. We recall here the formulation of the Monge–Kantorovich problem for smooth densities and domains in Euclidean space. For more general measures, see [1]. Let Ω_0 and Ω_1 be two diffeomorphic connected subdomains of $\mathbb{R}^d$, and let μ_0, μ_1 be Borel measures on Ω_0 and Ω_1, each with a strictly positive density function $\mu_0(x) \geq \mu_{\text{low}}^0 > 0$ and $\mu_1 \geq \mu_{\text{low}}^1 > 0$, respectively. Assume

$$\mu_0(\Omega_0) = \mu_1(\Omega_1),$$

i.e.,

$$\int_{\Omega_0} \mu_0(x)dx = \int_{\Omega_0} \mu_1(x)dx,$$

so that the same total mass is associated with Ω_0 and Ω_1.

The version of Monge-Kantorovich problem of interest in this work may be written

as follows:

$$\min \qquad M(u) := \frac{1}{2}\int_{\Omega} \mu_0(x)\rho(u,x)\,dx \tag{2.1.1a}$$

$$\text{s.t.} \qquad c(u) = \det(\nabla u)\mu_1(u(x)) - \mu_0(x) = 0, \tag{2.1.1b}$$

where u is a $C^{1,\alpha}$ diffeomorphism from $\Omega_0 \to \Omega_1$ and $\rho(u,x)$ is a distance function between x and u. Here we only treat the case

$$\rho(u,x) = |u-x|^2.$$

The constraint $c(u) = 0$ (the Jacobian equation) is often referred to as the mass-preserving (MP) property.

Even with a simple, quadratic distance function, this is a highly nonlinear equality constrained optimization problem. There is extensive analysis as for the existence, uniqueness, and properties of the solution (see for example [1, 33, 94] and the references therein). However, while there is a large body of literature that deals with the analysis of the problem, there is a surprisingly small number of papers that deal with the solution of the problem, and even a smaller number of papers that deal with efficient *numerical* solutions of the problem [2, 11, 22, 28, 75].

Among the papers that deal with the numerical solutions are the paper of Benamou and Brenier [11]. Their paper reconstructs an optimal path from μ_0 to μ_1 by solving an optimization problem with a space-time transport partial differential equation as a constraint. Their approach is particularly useful if the transportation path is needed. An interesting geometric method has also been formulated by Cullen and Purser [27].

A different approach that is the starting point of this work is presented by Angenent, Haker, and Tannenbaum (AHT) [2]. This approach reconstructs the transformation directly. The idea of the AHT method is to obtain an initial mass preserving

transformation $u_0(x)$, and then replace the original problem with the following optimization problem:

$$\min \qquad M(s) = \frac{1}{2}\int_\Omega \mu_0(x)|u_0(s^{-1}) - x|^2\, dx \tag{2.1.2a}$$

$$\text{s.t.} \qquad c(s) = \det(\nabla s)\mu_0(s(x)) - \mu_0(x) = 0, \tag{2.1.2b}$$

where $s \in \mathcal{C}^{1,\alpha}$ is a mass preserving mapping from μ_0 to itself. It is well known that the optimization problems (2.1.1) and (2.1.2) are equivalent [66]. The iteration starts with $s = x$ and since a composition of mass preserving maps is also mass preserving, the authors assume that the MP constraint is valid throughout the optimization process. This allows them to obtain a time dependent partial differential equation, equivalent to a steepest descent flow, that converge (in functional space) to the solution of the problem. To implement their method in a discrete setting the authors use forward Euler for time stepping and a first order finite difference discretization of the spatial derivatives. What is essentially done in this methodology is to construct the so-called *polar factorization* [15, 35, 36, 66] of the initial mass-preserving mapping u_0, thus obtaining $u = \nabla\phi$ where ϕ is some scalar function.

There are three main shortcomings to their numerical approach. First, a robust method to obtain an initial MP mapping is needed. The authors have used 1D integration that leads to a MP mapping that is highly irregular. See also [71] for an approach to constructing the initial MP mapping. Thus, the solution of the optimization problem may be far from the initial point and as a result many iterations are needed to convergence. Second and more importantly, although a composition of MP maps is also MP, this is only correct in the continuous setting. In the *discrete* setting this assumption is incorrect. Since the MP constraint is never specifically enforced during the *discrete* AHT minimization process it "drifts" that is, the final mapping may not be mass preserving and therefore the discrete PDE may not converge to a discrete approximation of the problem. Finally, since the steepest descent is slowly

converging, thousands of iterations are needed to converge to an optimal solution. This is particularly demanding in 3D.

In this report we discuss a combination of techniques that do not suffer from the aforementioned difficulties. In particular, our approach obtains a "reasonable" initial point that favors the solution of the problem by using nonlinear projection to a modified objective function. The same projection can be used as secondary corrections to the iterates that the AHT methods produces. Finally, we show that it is possible to modify the AHT approach and allow for directions other than steepest descent.

2.2 The AHT Method

We now review the derivation of the AHT method [2, 48]. In both papers the authors have used time embedding to derive their formulation. While time embedding can help to analyze the flow, it does not in general lead to efficient descent procedure [5]. In this work, we propose a variational approach that is more natural for the minimization of the L^2 Monge-Kantorovich functional rather than the gradient flow.

The key idea is as follows. Assuming that s is on the MP constraint manifold (2.1.2b), we perturb s by a small function $\delta s \in \mathcal{C}^{1,\alpha}$, and require that the resulting function stay on the manifold. This leads to

$$\begin{aligned} 0 &= c(s+\delta s) - c(s) = \det(\nabla(s+\delta s))\mu_0(s+\delta s) - \det(\nabla s)\mu_0(s) \\ &= \det(\nabla s)\,(\nabla\cdot\,(\delta s(s^{-1}))(s))\mu_0(s) \;+\; \det(\nabla s)\,\nabla\mu_0(s)\cdot\delta s. \end{aligned}$$

This expression can be simplified as long as the constraint is valid. Since $\det(\nabla u) > 0$, we can divide, and rearranging we have

$$\begin{aligned} 0 &= (\mu_0\nabla\cdot\,(\delta s(s^{-1})))(s) \;+\; \nabla\mu_0(s)\cdot\delta s \\ &= \mu_0\nabla\cdot\,(\delta s(s^{-1})) \;+\; \nabla\mu_0\cdot\delta s(s^{-1}) = \nabla\cdot\,(\mu_0\delta s(s^{-1}). \end{aligned}$$

Defining $\delta\zeta := \mu_0 \delta s(s^{-1})$, we see that

$$\nabla \cdot \delta\zeta = 0.$$

Next, looking at $u = u_0(s^{-1})$, we can write $u(s) = u_0$ that implies

$$(\nabla u(s))\delta s + \delta u(s) = 0$$

or

$$\delta u = -(\nabla u)\delta s(s^{-1}).$$

Using the definition of $\delta\zeta$, we obtain (as long as the constraint is valid) the following:

$$\delta u = -\mu_0^{-1}(\nabla u)\delta\zeta \tag{2.2.1a}$$

$$0 = \nabla \cdot \delta\zeta. \tag{2.2.1b}$$

A similar calculation shows that

$$\delta M = \int_\Omega u \cdot \delta\zeta \, dx. \tag{2.2.2}$$

In [2, 48], the authors propose the use of the Helmholtz decomposition in order to obtain a descent direction. Here we use a different approach. First, we note that the divergence constraint can be eliminated by selecting

$$\delta\zeta = \nabla \times \delta\eta,$$

and thus to reduce the objective function M, we need to obtain a direction that yields a negative δM. In other words, we seek a direction $\delta\eta$ such that

$$\delta M = \int_\Omega u \cdot (\nabla \times \delta\eta) \, dx < 0.$$

Using the divergence theorem, we obtain that

$$\int_\Omega u \cdot (\nabla \times \delta\eta) \, dx = \int_\Omega (\nabla \times u) \cdot (\delta\eta) \, dx + \int_{\partial\Omega} u \cdot (\delta\eta \times \vec{n}) \, dx,$$

and therefore, the steepest descent direction is given by

$$\delta\eta = -\begin{cases} \nabla \times u & \delta\eta \in \Omega \\ u \times \vec{n} & \delta\eta \in \partial\Omega \end{cases}$$

This leads to the update

$$\delta\zeta = -\nabla \times \nabla \times u,$$

and finally to the steepest descent direction in u given by

$$\delta u = \frac{1}{\mu_0}(\nabla u)\nabla \times \nabla \times u. \tag{2.2.3}$$

We can now further analyze the behavior of the system. The elliptic operator $\nabla \times \nabla\times$ is a positive semidefinite operator, and so the iteration is stable as long as

$$Re\,(\nabla u) \succ 0.$$

If at some point, one or more of the eigenvalues of ∇u become negative, then we obtain an ill-posed process. This crucial point must be treated within the chosen numerical method.

Using the above decomposition, a family of different directions may be obtained. Note that in order to reduce the objective $\int_\Omega \nabla \times u \cdot \eta \, dx$ any vector field of the form

$$\delta\eta = -A\nabla \times u$$

may be used. For example, a choice that leads to a similar method to the one derived in the papers [2, 48] in 2D is $A := -\Delta^{-1}$ that leads to the update

$$\delta u = -\frac{1}{\mu_0}(\nabla u)\nabla \times \Delta^{-1}\nabla \times u. \tag{2.2.4}$$

Using the above calculation, it is easy to see that the flow (2.2.4) is valid also in 3D.

Clearly, the formulation (2.2.4) is better scaled than (2.2.3). This is because the operator $\nabla \times \Delta^{-1}\nabla\times$ is compact while the $\nabla \times \nabla\times$ operator is unbounded. Thus, while the formulation (2.2.3) should have different convergence properties to different

frequencies the formulation (2.2.4) will not in general prefer high or low frequencies. The system (2.2.4) is nonlocal due to the term Δ^{-1}. We introduce a new variable $\delta p = \Delta^{-1}\nabla \times u$ and rewrite (2.2.4) in a local form

$$\begin{pmatrix} 1 & \mu_0^{-1}(\nabla u)\,\nabla\times \\ 0 & \Delta \end{pmatrix} \begin{pmatrix} \delta u \\ \delta p \end{pmatrix} = \begin{pmatrix} 0 \\ \nabla \times u \end{pmatrix} \tag{2.2.5}$$

In [2], the authors have proved that the proposed method is *globally convergent*. This is a major attraction of their technique. Nevertheless, convergence can be obtained only if very small steps are taken. After a descent direction is obtained, one can update the solution $u \leftarrow u + \alpha\delta u$ where $\alpha > 0$ guarantees reduction of the objective function. Assume that such an iteration is obtained, that is, set $w = u + \alpha\delta u$. The calculation above assumed that $\alpha\delta u$ was infinitesimal. However, unless $\alpha \to 0$, this assumption will be violated. Therefore, although u is MP, there is no reason to believe that w is also MP. Hence, a process of secondary correction is needed in order to maintain a MP transformation. The same process can be used to initialize u and obtain the initial MP transformation u_0. Since the entire scheme depends upon finding such a mapping, developing a method to obtain a good initial MP transformation is very important. We now explore this process in detail.

2.3 Obtaining a mass-preserving transformation

A key step in the AHT method is the initialization of the MP transformation and the projection to the MP constraint. In this section, we discuss this process.

Assume that we have a transformation w that is not mass preserving, and that we would like to obtain a transformation u that is mass preserving. This can be done by projection, that is, by solving the optimization problem

$$\min_u \quad \|u - w\|_{\mathcal{H}}^2 \tag{2.3.1}$$

$$\text{s.t} \quad c(u) = 0, \tag{2.3.2}$$

where $\mathcal{H}$ is some appropriate norm that we discuss next. There are many techniques to solve constrained optimization problems. In the following, we will consider a version of sequential quadratic programming (SQP) approach that is commonly used methodology for constrained optimization problems [73]. The advantage of SQP type methods is that they are mesh-independent. That is, the number of iterations to convergence is independent of the mesh size [95].

Note that for the MK problem one must consider a projection from the given point w with respect to the μ_0-weighted $L2$ norm $\|\cdot\|_{\mu_0}$ (see (2.1.1) above). Thus it should be clear that while we should be able to obtain a *local* minimizer for (2.3.1), it may not be the global one. Nevertheless, while we cannot guarantee *global* convergence of the projection process we can increase the chances that the local minimizer of (2.3.1) is also a global minimizer of the MK problem. We recall that the global minimizer has the property that $u = \nabla\phi$ [1, 94] and therefore, $\nabla \times u = 0$. Thus we propose to perform the projection by solving the following optimization problem:

$$\textbf{PROJ} \quad \min_{u} \quad \frac{1}{2}\int_{\Omega} \left(\mu_0(x)|u-w|^2 + \beta|\nabla \times u|^2\right)\, dx \tag{2.3.3a}$$

$$\text{s.t} \quad c(u) = 0, \tag{2.3.3b}$$

where $\beta > 0$. The term $\beta|\nabla \times u|^2$ does not change the global minimum. However, when solving for the projection it gives a bias towards a curl-free solutions, and thus can yield excellent starting points for the AHT method.

One can check that the conditions for a minimum lead to the following nonlinear system of equations:

$$\mu_0(u-w) \;+\; \beta\nabla \times \nabla \times u + \ldots$$

$$p\det(\nabla u)\,\nabla\mu_0(u) \;-\; \nabla\cdot\left(\det(\nabla u)\,(\nabla u)^{-\top}p\right) = 0 \tag{2.3.4a}$$

$$\det(\nabla u)\mu_0(u) = 0. \tag{2.3.4b}$$

There is an important practical difference between the AHT method and the projection step. Note that in the AHT method no interpolation, $\mu(u)$, is calculated.

This has the advantage of simplicity. However, as discussed in the previous section, since $\mu(u)$ is never calculated the MP constraint may not be enforced at the end of the process.

The optimization problem **PROJ** is a continuous constrained optimization problem. There are many techniques to obtain the solution to such problems. In the following, we will consider a version of sequential quadratic programming (SQP) which is a commonly used methodology for such constrained optimization problems [73]. The advantage of SQP type methods is that they are mesh-independent. That is, the number of iterations to convergence is independent of the mesh size [95].

In order to solve for the projection step two different possible approaches can be used. In the "discretize-then-optimize" approach, we discretize the projection step (2.3.3), and then solve a discrete optimization problem. In the "optimize-then-discretize" approach, we first write the Euler-Lagrange equations (2.3.4) in continuous setting, and then discretize them in order to obtain a system of nonlinear equations. The two approaches are summarized in [40]. The advantage of the discretize-then-optimize approach is that common optimization algorithms can be used with little modification. This is particularly important if we wish to obtain convergence to a minimum from a far-away starting point. Recent work in the field [73, 20] suggest several mechanisms to achieve this goal. We explore this issue in some detail below. Particular care must be taken so that the appropriate inner products and norms are discretized appropriately [50]. We therefore proceed with the discretization of the optimization problem.

We should note that although we use the discretize-then-optimize approach, the discrete nonlinear system and its linearization can be thought of as a discretization of the Euler-Lagrange equations (2.3.4). As such we need efficient solvers to systems that involve the **curl** operator and the linearization of the constraints.

2.4 Discretization

In order to discretize the problem, we define the Lagrangian

$$\mathcal{L}(u,p) = \frac{1}{2}\int_{\Omega} \left(\mu_0(x)|u-x|^2 + \beta|\nabla \times u|^2\right)\ dx + (p, c(u)) \tag{2.4.1}$$

where p is a Lagrange multiplier.

Certain key applications (for example in medical image processing) have data which are discretized on a regular grid. We therefore construct our discretization based on a finite volume/difference approach. Rather than working with the deformation field u, we work with the update or displacement $v = u - x$. This yields numerically smaller perturbations. Note also that $\nabla \times u = \nabla \times v$, and thus the projection (2.3.3) does not change.

We divide the domain, Ω, into $n_1 \times \ldots \times n_d$ cells, each of size $h_1 \times \ldots \times h_d$ where d is the dimension of the problem. For simplicity we discretize all the components of v at the nodes of each cell to obtain d grid functions $\hat{v}^1, \ldots \hat{v}^d$. The Lagrange multiplier p is discretized at the cell centers. In 2D, we denote by $\hat{v}^i_{j,k}$ the i^{th} grid function discretized at the $(x_1)_j, (x_2)_k$ node and by $\hat{p}_{j+\frac{1}{2},k+\frac{1}{2}}$ the Lagrange multiplier discretized at the cell center. The discretized quantities are plotted in Figure 1.

We note that a staggered discretization for the displacement can also be used [45]. Similar to problems that are derived from computational fluid dynamics and electromagnetics, such a discretization has very nice numerical properties. However, it is less simple to implement and requires careful treatment of boundary conditions. A discussion about the staggered versus the unstaggered discretization can be found in [93].

We now consider the different numerical approximations needed in order to obtain a discrete approximation to the continuous system. A consistent discretization for $\Delta, \nabla\times$ and a consistent discretization of $\nabla u = I + \nabla v$ are needed. There are a number of possible discretizations that lead to a well-posed system. Here we derive

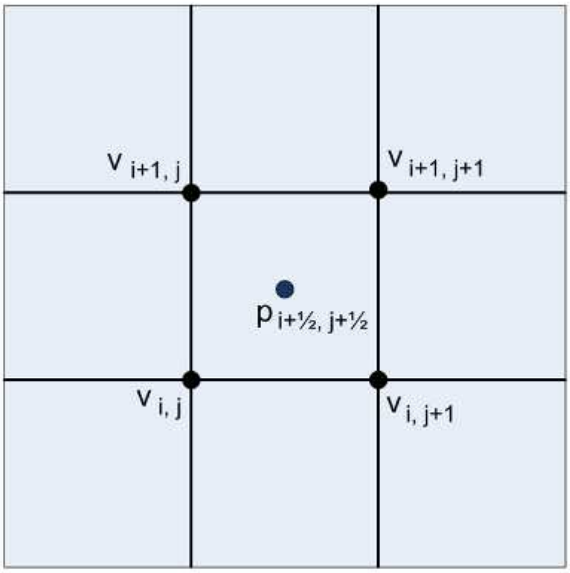


Figure 1: Discretization of the displacement and Lagrange multipliers.

the explicit discretization in 2D. The extension to 3D is straightforward.

2.4.1 Discretization of Δ and $\nabla\times$

In order to obtain a consistent discretization of the Laplacian, we use a standard discretization (5 point stencil in 2D and 7 point stencil in 3D) with Dirichlet boundary conditions.

To compute the discretization of the **curl** we use short differences in one direction averaged in the other direction. For example, in 2D we obtain

$$\begin{aligned}(C\hat{v})_{i+\frac{1}{2}+\frac{1}{2}} &= \frac{\hat{v}1_{i,j+1} - \hat{v}1_{i,j} + \hat{v}1_{i+1,j+1} - \hat{v}1_{i+1,j}}{2h_1} \\ &- \frac{\hat{v}2_{i+1,j} - \hat{v}2_{i,j} + \hat{v}2_{i+1,j+1} - \hat{v}2_{i,j+1}}{2h_2} + \mathcal{O}(h^2),\end{aligned} \tag{2.4.2}$$

where we denote by C the **curl** matrix. It is easy to verify the second order accuracy of this discretization.

2.4.2 Discretization of the Mass Preserving Constraint

In order to be able to perform the projection step, we need to discretize and solve the MP constraint. We use a finite volume discretization, and so integrating over a

given region Ω_j

$$\int_{\Omega_j} c(v)\,dx = \int_{\Omega_j} \det(I+\nabla v)\mu_0(x+v)\,dx$$

we see that two approximations are therefore needed. First, we need to approximate $\det(I+\nabla v)$ and second, we need to approximate $\mu_0(x+v)$.

There are a number of options for discretizing the constraint. In our application, μ_0 is typically discretized in cell centers and therefore we choose the domain Ω_j as a cell in our discretization.

To approximate $\mu_0(x+v)$ in cell centers we use linear interpolation; see [69] for details. We denote this approximation as $\mu_0(x_h+\hat{v})$ where x_h is the discretization of $x=[x_1,x_2]$ on the cell centers, $(i+\frac{1}{2}, j+\frac{1}{2})$. In order to approximate $\det(I+\nabla v)$ at the cell centers, we first compute all the derivatives in cell centers as follows:

$$(v^k_{x_1})_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{\hat{v}^k_{i+1j} - \hat{v}^k_{ij} + \hat{v}^k_{ij+1} - \hat{v}^k_{ij+1}}{2h} + \mathcal{O}(h^2) \tag{2.4.3a}$$

$$(v^k_{x_2})_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{\hat{v}^k_{ij+1} - \hat{v}^k_{ij} + \hat{v}^k_{i+1j+1} - \hat{v}^k_{i+1j}}{2h} + \mathcal{O}(h^2) \quad k=1,2. \tag{2.4.3b}$$

We also denote the block diagonal matrix

$$\mathbf{GRADv} = \text{blkdiag}\begin{pmatrix} (v^1_{x_1})_{i+\frac{1}{2},j+\frac{1}{2}} & (v^1_{x_2})_{i+\frac{1}{2},j+\frac{1}{2}} \\ (v^2_{x_1})_{i+\frac{1}{2},j+\frac{1}{2}} & (v^2_{x_2})_{i+\frac{1}{2},j+\frac{1}{2}} \end{pmatrix}. \quad i=1,\ldots,n_1{-}1,\ j=1,\ldots,n_2{-}1$$

This matrix is the discrete analog to the matrix ∇v at each cell center. For the MK problem to be well-defined, the matrix $I+\nabla v$ must be SPD everywhere.

Let D^s_j be the discretization of the j^{th} derivative. The discrete mass preserving constraint for v then reads

$$\begin{aligned}(\mathbf{1} + D_1\hat{v}_1 + D_2\hat{v}_2 + (D_1\hat{v}_1)\odot(D_2\hat{v}_2) - (D_2\hat{v}_1)\odot(D_2\hat{v}_1))\ldots \\ \odot\ \mu_0(x_h+\hat{v}) = \mu_1(x_h).\end{aligned} \tag{2.4.4}$$

where $\odot$ is the Hadamard product.

It is important to further study this discretization and its properties. Typically to any discretization we study the consistency and the stability of our discretization.

2.4.2.1 Consistency of the discretization

To study the consistency of the discretization, the we employ following lemma:

Lemma 1 *Let $\hat{v} = (\hat{v}^1, \ldots, \hat{v}^d)$ be grid functions of a sufficiently smooth field v, and let* **GRADv** *be the approximated for the matrix ∇v. Then, if $I + \nabla v$ has positive real eigenvalues with $Re(\lambda_{\min}) \geq \rho > 0$, the matrix* **GRADv** *also has positive real eigenvalues for a sufficiently small h.*

Proof 1 *To prove the lemma, we note that the matrix is a block 2×2 matrix. This implies that the eigenvalues of the matrix $I +$* **GRADv** *can be computed blockwise, where each of the blocks corresponds to a different cell in the discrete grid. Since the approximation of the derivatives is $\mathcal{O}(h^2)$, the eigenvalues of the matrix $I +$* **GRADv** *converge to the eigenvalues of $I + \nabla v$.*

It is important to note that the above lemma does not predict the order of convergence. Unfortunately, very small perturbations in the discrete grid function v can lead to large perturbations in the eigenvalues of **GRADv**. Consider for example, the case in which the approximation to **GRADv** at some cell is

$$A = I + \mathbf{GRADv} = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix},$$

and that we have a numerical error of the form ϵE where

$$E = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

Then, it is easy to verify that

$$\lambda(A + \epsilon E) = 1 \pm \sqrt{\epsilon a}.$$

Thus, for large enough a the eigenvalues of the perturbed system can be very different from the eigenvalues of the continuous one.

Nevertheless, the above lemma is important, because it implies that the overall (discrete) problem is consistent for sufficiently small h. It is also important because it gives certain hints about possible difficulties with the numerical solution. It is well-known that the continuous problem can be ill-posed if the real parts of the eigenvalues of ∇u are not positive, that is, $\mathrm{Re}(\nabla u) \succ 0$ (see [33]). The lemma requires h to be sufficiently small in order to obtain this behavior. If the displacement field is not sufficiently smooth or h is not sufficiently small, then we may encounter negative eigenvalues in the discrete approximation of u. In this case the iteration can stall or even diverge. We therefore propose to check the eigenvalues of $I + \mathbf{GRADv}$ throughout the iteration. This can be done in $\mathcal{O}(n)$ work due to the block structure of the system.

There are a number of ways to correct for negative eigenvalues. Denote the Schur decomposition of the gradient matrix at the $i + \frac{1}{2}, j + \frac{1}{2}$ cell by

$$(I + \mathbf{GRADv})_{i+\frac{1}{2},j+\frac{1}{2}} = V_{i+\frac{1}{2},j+\frac{1}{2}} \Lambda_{i+\frac{1}{2},j+\frac{1}{2}} V^{\top}_{i+\frac{1}{2},j+\frac{1}{2}}$$

where all the matrices are 2×2 (in 2D). For the sake of concreteness, assume that the first eigenvalue $\Lambda^1_{i+\frac{1}{2},j+\frac{1}{2}} \leq 0$. A simple modification of our method, is simply to inflate Λ^1_{ij} in such a manner that it is slightly larger than 0, for example, to h. We have taken precisely this approach to avoid vanishing or negative eigenvalues that may lead to divergence of the method.

2.4.2.2 Stability of the discretization

Although the discretization of the MP constraint is consistent and second order accurate, it is *not* a stable discretization. It is easy to verify that the derivatives have a non-trivial null space known as a *checkerboard null-space*. This is not very surprising. Indeed, consider the simple case of a uniform mass in which one linearizes the perturbation around it. The Taylor expansion yields

$$\det(I + \nabla \delta v) \approx 1 + \nabla \cdot \delta v.$$

Thus, for the starting point $u = x$ and $v = 0$ the first iterate gives a system similar to the Stokes system. It is well-known that non-staggered discretizations of the Stokes equation are not stable [93]. Nevertheless, simple stabilization techniques can be used to obtain a well-posed discrete system. The most common approach is a penalty method [93] where an artificial viscosity is added to the Lagrange multiplier. This additional term stabilizes the system without affecting the overall accuracy of the problem. We return to this issue in the next section.

2.4.3 Discretization of the Objective Function

In order to discretize the objective function on the regular grid, two quantities are needed. First, we need to approximate $\int \mu_0 |v|^2 \, dx$ on Ω_j, and second the approximation of $\int |\nabla \times \, v|^2 \, dx$ is needed.

Assuming μ_0 is given in cell-centers we obtain

$$\int_{\Omega_k} \mu_0 |v|^2 dx = h^2 \frac{1}{4} \mu_{0,(i+\frac{1}{2},j+\frac{1}{2})} \sum_{\ell=1}^{2} (\hat{v}_{ij}^{\ell} + \hat{v}_{i+1j}^{\ell} + \hat{v}_{ij+1}^{\ell} + \hat{v}_{i+1j+1}^{\ell})^2 + \mathcal{O}(h^2). \quad (2.4.5)$$

Summing over the cells we obtain a second order accurate discretization that can be written in matrix form as

$$\int_{\Omega} \mu_0 |v|^2 dx = \hat{v}^{\top} M_{\mu_0} \hat{v} + \mathcal{O}(h^2), \quad (2.4.6)$$

where M_{μ_0} is a μ_0 weighted mass matrix.

Since the discretization of the **curl** is centered at the cell centers it is straightforward to use the **curl** matrix C to obtain

$$\int_{\Omega} |\nabla \times \, v|^2 dx = h^2 \hat{v}^{\top} C^{\top} C \hat{v} + \mathcal{O}(h^2). \quad (2.4.7)$$

2.5 Computation of a Step

The computation of a step requires two parts. First, we need to verify that the constraint is feasible. This is done by computing the projection step. After the constraint is satisfied, we take a step using the AHT method. We now discuss each of these steps and the algorithm that results from their combination.

2.5.1 Computation of the AHT Step

Using the discretization of the operators we now have a discrete analog of (2.2.5) that we write as

$$\begin{pmatrix} I & M_{\mu_0^{-1}}(I + (\nabla_h \hat{v}))\, C^\top \\ 0 & \Delta_h \end{pmatrix} \begin{pmatrix} \delta\hat{v} \\ \delta\hat{p} \end{pmatrix} = \begin{pmatrix} 0 \\ C\hat{v} \end{pmatrix} \tag{2.5.1}$$

Here $M_{\mu^{-1}}$ is the mass matrix of μ^{-1} obtained in a similar way to the mass matrix of μ, M_μ. The solution of the system (2.5.1) is straightforward. Any fast Poisson solver can be used for the task. In this work, we have used a standard multigrid method with weighted Jacobi smoothing, bilinear prolongation and its adjoint as a restriction [93]. The details of the computation of the AHT step using Multigrid methods are provided in the next chapter alongwith the details of using a commodity Graphics Processor Unit (GPU) to compute in near real-time speeds.

After the step is computed we need to decide if it is an acceptable step. This is a rather delicate point. The AHT method is an *unconstrained optimization* technique, that is, it assumes that the MP constraints have been eliminated. For unconstrained optimization techniques, a sufficient reduction in the objective function is a criterion to accept the step [73]. The problem is that a simple update of the form $\hat{v} \leftarrow \hat{v} + \alpha\delta\hat{v}$ is not mass preserving in general even if $\hat{v}$ is. This is because the elimination of the MP constraint is done for the *linearized* equations, and therefore unless α is infinitesimally small, the updated solution is not MP. This implies that the magnitude of the objective function before and after the update is not a valid measure to step acceptance. Consequently, care must be taken in order to guarantee that the MP constraint is enforced after each step of the AHT method. This is similar to secondary correction methods and projection to the central path that is often done in constrained optimization algorithms [73]. Assume that at iteration n, we have $\hat{v}_n$ as an approximation to v and that we computed $\delta\hat{v}$. To update we use the following

formula

$$\hat{v}_{n+1} = \mathcal{P}(\hat{v}_n + \alpha\delta\hat{v}), \tag{2.5.2}$$

where $\mathcal{P}$ is the nonlinear projection discussed in Section 2.3. The nonlinear projection operator $\mathcal{P}$ projects $\hat{v}_n + \alpha\delta\hat{v}$ into the MP manifold. The step size α is then chosen such that the objective function of the projected step is decreased. We summarize this scheme in Algorithm 1.

Algorithm 1 Solution of OMT:
$\hat{v} \leftarrow \text{OMTsol}(\mu_0, \mu_1)$;

```
Use μ_0 and μ_1 to compute a mass preserving v̂_0 = P(0)
while true do
  Compute ∇_h v̂ and correct such that I + Re(∇_h v̂) ≻ 0
  Solve (2.2.5) for δv̂
  line search: set α = 1
  while true do
    v̂_{n+1} = P(v̂_n + αδv̂)
    if ‖v̂_{n+1}‖_{μ_0} < ‖v̂_n‖_{μ_0} then
      Break
    end if
    α ⟸ α/2
  end while
end while
```

2.5.2 Computation of the Projection Step

The main building block of our algorithm is the projection to the MP constraint (2.3.3). The projection enables us to initialize the AHT method as well as the projection of any AHT step back to the MP constraint.

Using the discretization of the objective function and the constraint we obtain the following discrete optimization problem:

$$\min_{v} \quad \frac{1}{2}\hat{v}^\top M_{\mu_0}\hat{v} + \frac{\beta}{2}\|C\hat{v}\|^2 \tag{2.5.3a}$$

$$\text{s.t} \quad (\mathbf{1} + D_1\hat{v}_1 + D_2\hat{v}_2 + (D_1\hat{v}_1) \odot (D_2\hat{v}_2) - (D_2\hat{v}_1) \odot (D_2\hat{v}_1)) \odot \dots$$

$$\mu_0(\hat{v}) = \mu_1. \tag{2.5.3b}$$

Given the properties of our discretization we now discuss the a modification of the Lagrangian and the computation of a step.

In order to overcome the problem of stability in the discrete constraint, we use a well-known strategy utilized in computational fluid dynamics [93] and add a small penalty term to the discrete Lagrangian, defined as

$$\mathcal{L} = \frac{1}{2}\hat{v}^\top M_{\mu_0}\hat{v} + \frac{\beta}{2}\hat{v}^\top C^\top C\hat{v} + p^\top c(v) - \frac{\gamma}{2}\|\nabla_h \hat{p}\|^2 \tag{2.5.4}$$

where γ is an $\mathcal{O}(h^2)$ parameter. The latter term is added to the problem to overcome the stability problem. For a simple constraint $c(v) = \nabla \cdot v$ it is possible to obtain an analytic expression for the optimal γ (see, for example, [93]). However, when the constraint depends on v such optimal expression is not attainable. We therefore set $\gamma = h_1 h_2$ which now yields a consistent and stable approximation to the Lagrangian.

To solve the problem, we use a version of inexact Sequential Quadratic Programming (SQP) [20]. The bottleneck in this methodology is the solution of the so called KKT system

$$\begin{pmatrix} H & c_u^\top \\ c_u & -S \end{pmatrix} \begin{pmatrix} \delta\hat{v} \\ \delta\hat{p} \end{pmatrix} = \text{RHS}, \tag{2.5.5}$$

where H is an approximation to the second derivatives of the Lagrangian and c_u is the Jacobian of the constraint. Here we used $H = M_\mu + \beta C^\top C$. This approximation to the Hessian yields a SPD approximation to the (1,1) block and therefore guarantees a descent direction. We have modified standard SQP algorithms and introduced the regularization matrix $S = h_1 h_2 \Delta_h \succ 0$ in our numerical procedure.

To solve the KKT system, we have taken a number of steps. First, we have used an augmented Lagrangian approach [37] and generated the modified system

$$\begin{pmatrix} H_{ag} & c_u^\top \\ c_u & -S \end{pmatrix} \tag{2.5.6}$$

where $H_{ag} = H + c_u^\top c_u$. The augmentation helps to deal with the null space of the **curl** operator and is commonly used in computational electromagnetics [42, 39].

Next we use a block preconditioner similar to the one discussed in [86]. It is well-known that an optimal preconditioner for the KKT system is [12]

$$\begin{pmatrix} H_{ag} & c_u^\top \\ 0 & S_c \end{pmatrix} \tag{2.5.7}$$

where S_c is the Schur complement

$$S_c = c_u H_{ag}^{-1} c_u^\top + S.$$

The matrix H_{ag} is derived from an elliptic operator, and therefore it is possible to approximate its inverse using standard multigrid methods. To approximate the Schur complement matrix we drop the matrix S which is $\mathcal{O}(h^2)$ and use the pseudo-inverse of c_u, to obtain

$$S_c^{-1} \approx c_u^\dagger H_{ag} c_u^\ddagger.$$

The computation of $c^\dagger$ times a vector z requires the solution of a system of the form

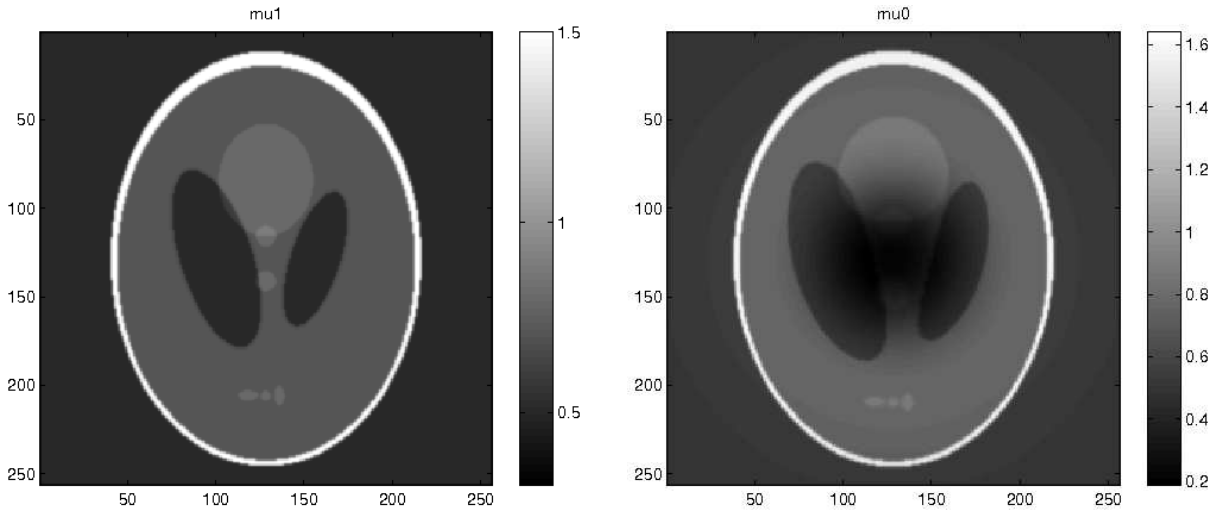
$$c_u c_u^\top q = z.$$

Since $c_u c_u^\top$ is a discretization of an elliptic operator this can be done using a standard multigrid technique. The numerical properties and performance of this approximation will be studied elsewhere. In our numerical experiments we have noticed that the number of iterations of the generalized minimal residual method (GMRES) [84] that was needed to solve the KKT system was almost mesh independent.

2.6 Numerical Experiments

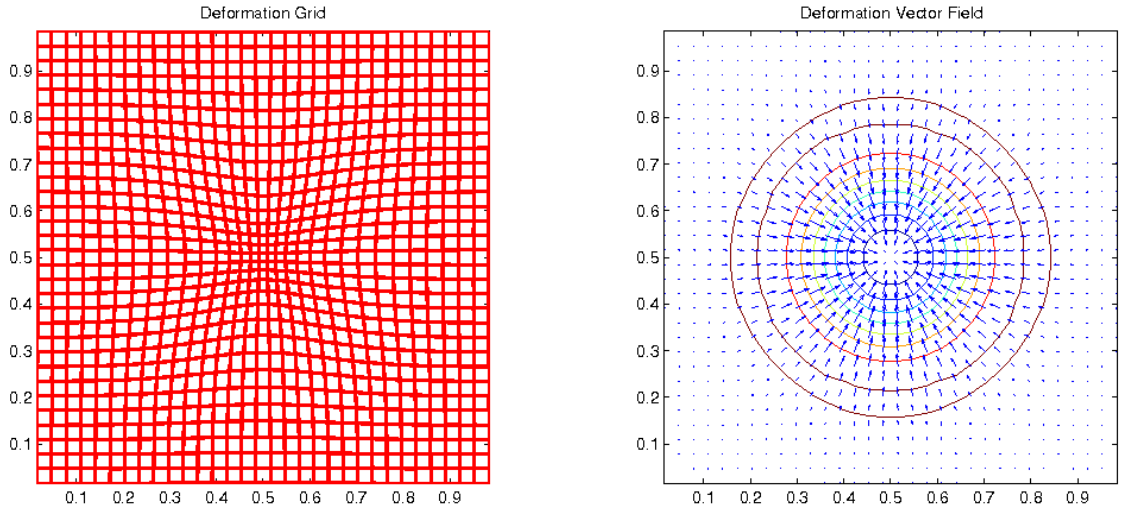
Since the solution of the problem is also the solution of the Monge-Ampere equation [33] it is easy to construct an analytic example. In order to do this, we consider the following function

$$\phi(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2) + c \cdot e^{-\frac{1}{2}(x_1 - \frac{1}{2})^2/\sigma_1^2} \cdot e^{-\frac{1}{2}(x_2 - \frac{1}{2})^2/\sigma_2^2} \in [0, 1]^2,$$



(a) MATLAB Phantom Image (μ_1)

(b) Deformed Image (μ_0)



(c) Deformed Grid

(d) Deformation Vector Field

Figure 2: 2D Analytical Example. Two image pairs (a) & (b) with a known deformation, mapping one to the other. (c) shows the deformed grid and (d) shows the deformation vector field with superimposed gradient contours of the determinant of the Jacobian.

where c, σ_1 and σ_2 are parameters chosen to create a unique deformation field. Differentiating ϕ with respect to $x = (x_1, x_2)$, we obtain $u = (u_1, u_2)$,

$$\begin{aligned} u_1 &= x_1 - c \cdot ((x_1 - 0.5)/\sigma_1^2) \cdot e^{-\frac{1}{2}(x_1-\frac{1}{2})^2/\sigma_1^2} \cdot e^{-\frac{1}{2}(x_2-\frac{1}{2})^2/\sigma_2^2} \\ u_2 &= x_2 + c \cdot ((x_2 - 0.5)/\sigma_2^2) \cdot e^{-\frac{1}{2}(x_1-\frac{1}{2})^2/\sigma_1^2} \cdot e^{-\frac{1}{2}(x_2-\frac{1}{2})^2/\sigma_2^2}). \end{aligned}$$

We employed a standard 2D phantom image from MATLAB to serve as $\mu_1(x_1, x_2)$ and define $\mu_0(x_1, x_2)$ by

$$\mu_0 := \det(\nabla u)\mu_1(u). \tag{2.6.1}$$

We computed $\mu_0(x_1, x_2)$ via linear interpolation. The deformed grid, deformation vector field, μ_0 and μ_1 thus obtained are shown in Figure 1. We next inputed the μ_1 and μ_0 obtained in this manner into our solver to find the transformation u using our algorithm.

We also performed the same experiment for a 3D brain MRI dataset. The deformation field from the 2D example was symmetrically extended in the third dimension and the deformed image was computed in the same way. For both experiments we use $\beta = 10^2$. We terminated our algorithm after 100 iterations or when the curl of the solution was 4 orders of magnitudes smaller than its initial size (in the infinity norm).

Table 1 shows the ∞-norm of the error between the known and compute deformation fields at different grid sizes.

The tables clearly demonstrate quadratic convergence of our method to the true solution that is expected from the discretization error used in our numerical approximations.

To demonstrate the efficiency of our algorithm we present in Table 2 the number of projection steps and AHT steps used for each of the examples on each level. Note that we have not used a multilevel procedure that could speed up calculations. This was done in order to demonstrate mesh-independence of the method. The incorporation

Table 1: Convergence for the 2D and 3D experiments

Grid size	Error (infinity norm) in 2D	Error (infinity norm) in 3D
2^{-3}	1.1×10^{-4}	1.2×10^{-2}
2^{-4}	2.5×10^{-5}	3.8×10^{-3}
2^{-5}	6.0×10^{-6}	9.0×10^{-4}
2^{-6}	2.4×10^{-7}	2.2×10^{-5}
2^{-7}	5.9×10^{-8}	5.1×10^{-6}
2^{-8}	1.2×10^{-8}	

Table 2: Number of projection and AHT iterations for the 2 and 3D experiments

Grid size	# proj iter (2D)	AHT iter (2D)	# proj iter (3D)	AHT iter (3D)
2^{-3}	56	3	67	3
2^{-4}	32	2	46	2
2^{-5}	28	1	34	1
2^{-6}	27	1	35	1
2^{-7}	28	1	36	1
2^{-8}	26	1	35	1

of a multilevel method will be done in future work. Table 2 clearly demonstrates the mesh independence property of the SQP algorithm. It also demonstrates that the modified projection is indeed very effective in obtaining an excellent starting point to the AHT method. Indeed, a single AHT iteration was performed by the code. This implies, the initial projected solution was at the solution of the MK problem.

CHAPTER III

MULTIGRID COMPUTATION OF AHT ON GRAPHICS PROCESSING UNITS

3.1 Introduction

Computationally, one of the attractive features of the AHT method of computing the optimal mass transport mapping is that the core of the algorithm is a Poisson solver. As we deal with larger and larger data sizes in our applications of OMT in image processing, computational efficiency becomes an important factor in determining the feasibility of the algorithm. We have tried to solve this issue by employing advanced numerical techniques such as Multigrid Methods, described in this chapter, and Adaptive Mesh Refinement using octrees described in Chapter 4. Moreover, the Multigrid methods employed are highly data-parallel computations in nature and are ideally suited for computations on newly emerging Graphics Processing Units (GPUs). We developed a library of functions that offload these numerical computations to the GPU resulting in remarkable overall speed-ups for our applications. In this chapter we provide the detail of these techniques.

3.2 Multigrid Methods

Multigrid methods were developed to solve boundary value problems on spatial domains [14]. These methods are particularly useful on sparse systems as a numerical method of solving equations in the category of iterative or relaxation methods, versus a direct method. Methods have been developed for both linear and nonlinear equation systems [56, 67, 90]. Direct methods solve the equations directly using a finite number of steps, such as through the cyclic reduction or fast Fourier transform. Relaxation

methods, such as the Gauss-Seidel or Jacobi methods, begin with an initial guess at a solution and proceed iteratively to the solution upon which it converges.

The multigrid idea is very fundamental. It takes advantage of the smoothing properties of the classical iteration methods at high frequencies (Jacobi, Gauss Seidel, SOR, etc.) and the error smoothing at low frequencies by restriction to coarse grids. The essential multigrid principle is to approximate the smooth (low frequency) part of the error on coarser grids. The non-smooth or rough part is reduced with a small number of iterations with a basic iterative method on the fine grid. In the discussion below, the methodology description of Briggs [16] is followed closely. We start with an overview of the traditional relaxation methods. Let,

$$Au = f \tag{3.2.1}$$

denote a system of linear equations, with u representing the exact solution of the system. Classical relaxation methods require computing time of $O(n^p)$, where n is the total number of grid points and p is usually 2 [54], therefore, these methods become computational unfeasible for large size problems. If v denotes the approximate solution to the exact solution u, then the error $e = u - v$ satisfies

$$Ae = r = f - Av \tag{3.2.2}$$

which says that we can relax directly on the error by using the residual equation. Moreover, relaxation on the original equation $Au = f$ with an arbitrary initial guess v is equivalent to relaxing on the residual equation $Ae = r$ with the specific initial guess $e = 0$. Then the approximation for the solution at next step is given is by $u = v + e$. This methodology of using residual corrections as a relaxation method is used in all classical relaxation methods such as Jacobi and Gauss-Seidel, just to name a few. These methods damp fluctuations in the residual, usually quickly at the beginning and then flatten out, a few iterations into the process and convergence becomes very slow. A useful feature of multigrid methods is that the convergence

rate is independent of the grid spacing. Therefore, when relaxation, or solution of the linear system has stalled on the finer grid, moving to a coarse grid allows one to now relax on the error by using the residual equation. By repeatedly relaxing on further coarser grids, this initial guess in improved each time. A two grid setup of this method would comprise of the following steps:

- Start on the fine grid and relax ν_1 times on $Au = f$ on the original fine grid to gain initial guess v.
- Compute the residual $r = f - Av$.
- Transfer residual to the coarse grid (restriction) and relax on $Ae = r$ to approximate the error.
- Transfer error to the fine grid (interpolation) and correct the approximate solution using $v = v_0 + e$.
- Relax ν_2 times on $Au = f$ using the corrected initial guess.

The idea of using coarser grids to generate improved initial guesses is the basis of a strategy called *nested iteration*. This two grid mechanism can be extended to a cascade of coarser grids such that the exact solution at the coarsest level can be obtained by a direct solution of the linear system and this mechanism is the basis for the development of different multigrid methodologies. The basic components of multigrid algorithm are discretization, inter-grid transfer operators (interpolation and restriction), a relaxation scheme and the iterative cycling structure. In the following section we describe our choice of these components for the 3D multigrid Poisson solver.

3.3 3D Multigrid Poisson Solver

A simple profiling of the code clearly shows that the most time consuming computation in the AHT algorithm is the solution of the Poisson equation. To obtain a fast

solver we employ a multigrid solver. We closely follow the approach proposed in [41]. For completeness we describe the key components of the algorithm here.

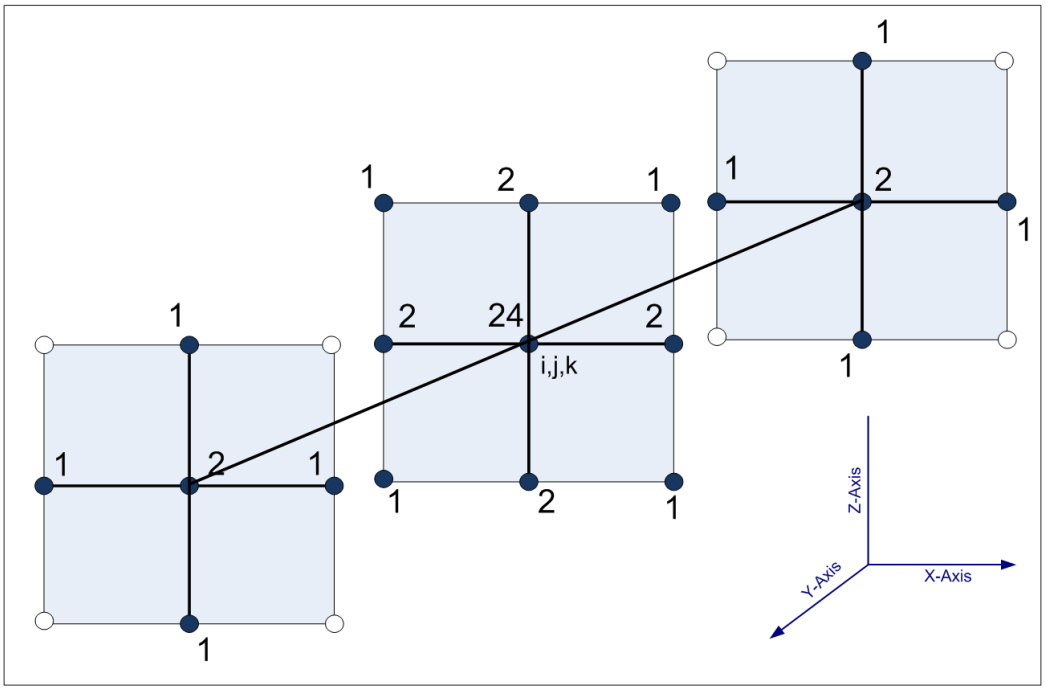


Figure 3: Stencil coefficients of the 19-point scheme for the 3D Poisson Equation

3.3.1 Discretization

The discrete Poisson equation is given as shown below

$$-\Delta u\left(x,y,z\right)=f\left(x,y,z\right),\left(x,y,z\right)\in\Omega, \tag{3.3.1}$$

where Ω is a continuous convex domain in 3D space with suitable boundary conditions prescribed on its boundary $\partial\Omega$. We employ the fourth order compact approximation scheme as proposed in [41]. The 3D stencil for this discretization is shown Figure 3 and the corresponding 19-point formula is as following:

$$
\begin{aligned}
&24u_{i,j,k} - 2\left(u_{i+1,j,k} + u_{i-1,j,k} + u_{i,j+1,k} + u_{u,j-1,k} + u_{i,j,k+1} + u_{i,j,k-1}\right) \\
&\quad - \left(u_{i+1,j+1,k} + u_{i-1,j+1,k} + u_{i+1,j-1,k} + u_{i-1,j-1,k} + u_{i+1,j,k+1} + u_{i-1,j,k+1}\right) \\
&\quad + \left(u_{i,j+1,k+1} + u_{i,j-1,k+1} + u_{i+1,j,k-1} + u_{i-1,j,k-1} + u_{i,j+1,k-1} + u_{i,j-1,k-1}\right) \\
&\quad = \frac{h^2}{2}\left(6f_{i,j,k} + f_{i+1,j,k} + f_{i-1,j,k} + f_{i,j+1,k} + f_{i,j-1,k} + f_{i,j,k+1} + f_{i,j,k-1}\right)
\end{aligned}
\tag{3.3.2}
$$

This 19-point formula has a truncation error of $\mathcal{O}\left(\langle^{\triangle}\right)$ and provides a computationally efficient approximation and uses less memory compared to a standard second order scheme.

3.3.2 Multigrid Cycling Structure

We use a standard V-cycle algorithm for iterative multigrid solver. One iteration of a simple multigrid V-cycle algorithm consists of smoothing the error using a relaxation technique (e.g., the Jacobi and GaussSeidel methods), solving an approximation to the smooth error equation on a coarse grid, interpolating the error correction to the fine grid, and finally adding the error correction into the current approximation. A multigrid V-cycle is the process that goes from the finest grid down to the coarsest grid and moves back from the coarsest up to the finest. A V(ν_1;ν_2)-cycle is a multigrid V-cycle algorithm that performs ν_1 relaxations on each level before projecting the residual to the coarse grid (pre-smoothing), and performs ν_1 relaxations after interpolating the coarse grid correction back to the fine grid (post-smoothing). The whole algorithm is shown in Algorithm 3.3.2 below. For other multigrid cycling algorithms, we refer to the books of Briggs, Wesseling and Trottenberg [16, 96, 93].

3.3.3 Inter-grid Transfer Operators

The prolongation operator I_{2h}^{h} for transferring coarse grid correction from coarse to fine grids is the trilinear interpolation. Specifically, correction values of common points of fine and coarse grids are directly transfered. Depending on the location,

Algorithm 2 Multigrid V-cycle Scheme:
$v^h \leftarrow MGV^h\left(v^h, f^h, \nu_1, \nu_2\right)$

Step1: Relax ν_1 times on $A^h u^h = f^h$ with a given initial guess v^h
if Ω^h = coarsest grid - (Step 2) **then**
  Goto Step 4
else
  $f^{2h} \leftarrow I_h^{2h}\left(f^h - A^h v^h\right)$
  $v^{2h} \leftarrow 0$
  $v^{2h} \leftarrow MGV^{2h}\left(v^{2h}, f^{2h}, \nu_1, \nu_2\right)$
end if
Step3: Correct $v^h \leftarrow v^h + I_{2h}^h v^{2h}$
Step4: Relax ν_1 times on $A^h u^h = f^h$ with a given initial guess v^h

correction values of other fine grid points are obtained by averaging correction values of the nearest two, four or eight points on the coarse grid. For detailed formulas, see page 67 in [96].

The residual restriction operator I_h^{2h} for projecting residual from fine to coarse grids is the full-weighting scheme. The full-weighting operator evaluates residuals at all fine grid points and projects a weighted average of residuals at the nearest 27 points in a cube centered at the reference point according to the following formula. For convenience, the stencil for the full-weighting operator is given in Figure 4.

$$\begin{aligned}
\bar{r}_{\bar{i},\bar{j},\bar{k}} =& \frac{1}{64}\left[8r_{i,j,k} + 4\left(r_{i+1,j,k} + r_{i-1,j,k} + r_{i,j+1,k} + r_{u,j-1,k} + r_{i,j,k+1} + r_{i,j,k-1}\right)\right. \\
&+ 2\left(r_{i+1,j+1,k} + r_{i-1,j+1,k} + r_{i+1,j-1,k} + r_{i-1,j-1,k}\right. \\
&+ r_{i+1,j,k+1} + r_{i-1,j,k+1} + r_{i,j+1,k+1} + r_{i,j-1,k+1} + r_{i+1,j,k-1} \\
&\left.+ r_{i-1,j,k-1} + r_{i,j+1,k-1} + r_{i,j-1,k-1}\right) \\
&+ r_{i+1,j+1,k+1} + r_{i-1,j+1,k+1} + r_{i+1,j-1,k+1} + r_{i-1,j-1,k+1} \\
&\left.+ r_{i+1,j+1,k-1} + r_{i-1,j+1,k-1} + r_{i+1,j-1,k-1} + r_{i-1,j-1,k-1}\right]
\end{aligned} \tag{3.3.3}$$

Here $r_{i,j,k}$ is the residual on the fine grid at a red point (i, j, k) and $\bar{r}_{\bar{i},\bar{j},\bar{k}}$ is the quantity to be transfered to the corresponding coarse grid point $\left(\bar{i}, \bar{j}, \bar{k}\right) = (i/2, j/2, k/2)$.

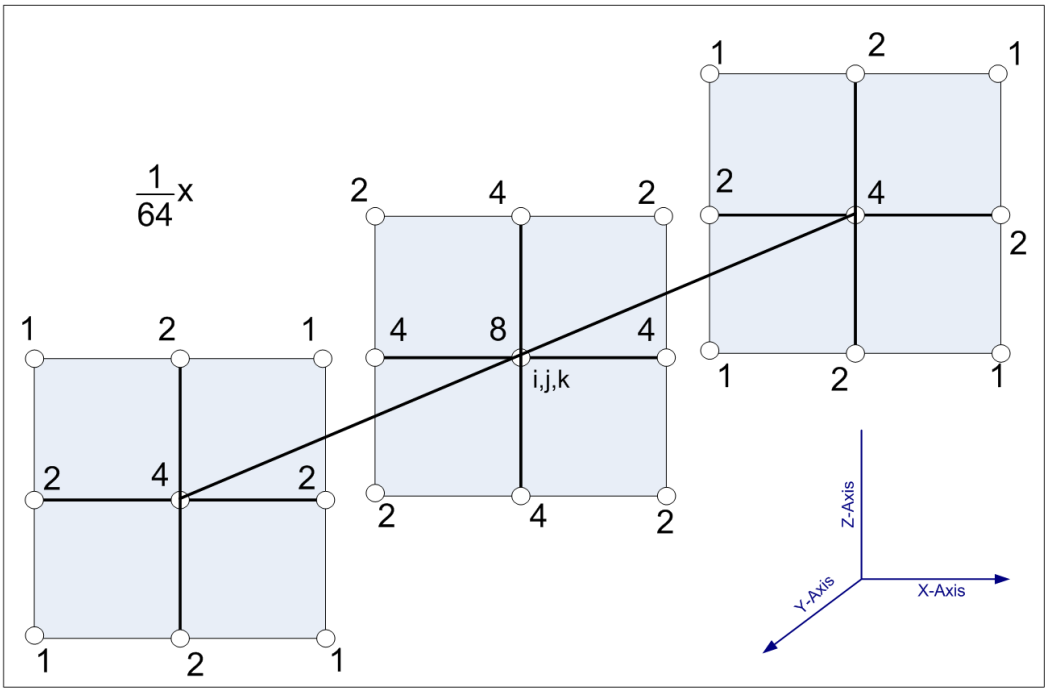


Figure 4: Stencil for the 3D Full-weighting Restriction Operator

3.3.4 Parallel Relaxation Using Colored Grids

We used a four-color Gauss-Seidel relaxation scheme in our multigrid algorithm. Using four colors allow to decouple the 3D grid points completely. The four color scheme ensures that for each grid point of a given color, the nearest grid points along the three coordinate directions are marked with different colors. Figure 5 shows this arrangement for a reference point colored with red and its 18 nearest neighboring grid points are colored with black, green and white. This scheme allows for all grid points with each color to be updated simultaneously on a parallel computer, a feature we exploit in our GPU implementation. Four sub-sweeps can be carried out to perform a Gauss-Seidel relaxation on the whole grid. This is known as the four-color Gauss-Seidel relaxation and in addition to parallelization and vectorization also provides better convergence and smoothing properties.

In this section we have outlined the multigrid algorithm employed for a parallel implementation and is in no ways an exhaustive introduction of these methods. The

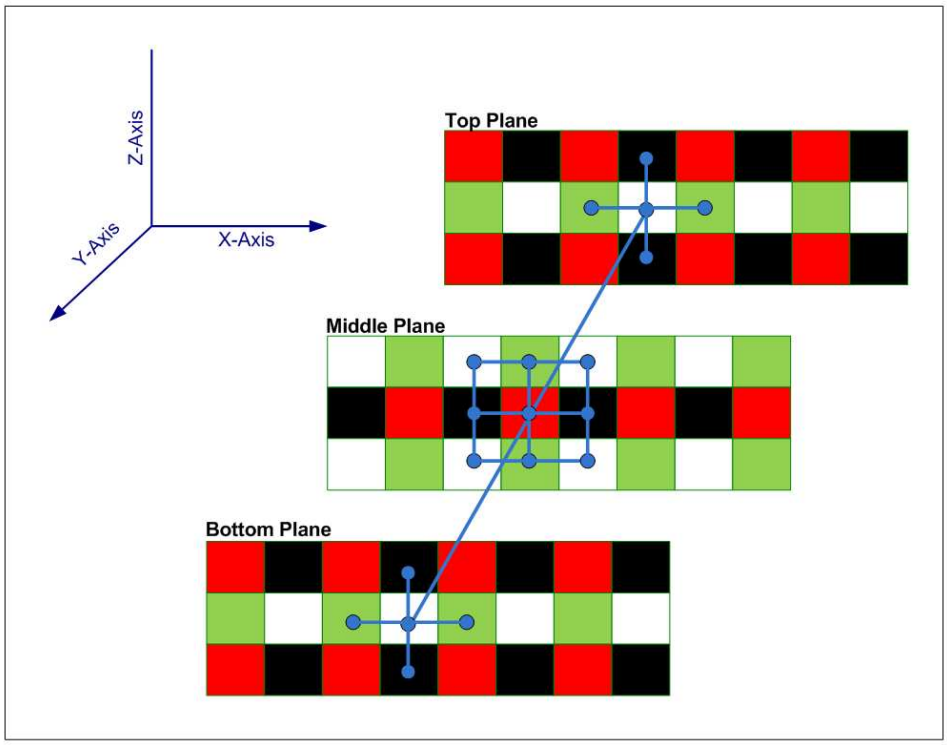


Figure 5: Decoupling the 3D grid points with four colors

interested reader is referred to [41, 16, 93, 96] for complete theoretical details on implementation of the multigrid method. In the following section we describe the parallel implementation of the AHT update step on a GPU.

3.4 GPU implementation

An advantage of our solution to the OMT problem is that it is particularly well-suited for implementation on parallel computing architectures. Over the past few years, it has been shown that graphics processing units (GPUs; now standard in most consumer-level computers), which are naturally massively parallel, are well suited for these types of parallelizable problems [13, 74]. In the following we provide an introduction to the model of performing general purpose computations on the GPU commonly known as the GPGPU model. We also provide the details of our implementation of the AHT update step on the GPU including the mulitgird Poisson solver.

3.4.1 Introduction to GPGPU

A GPU is a highly parallel computing device designed for the task of graphics rendering. However, the GPU has evolved in recent years to become a more general processor, allowing users to flexibly program certain aspects of the GPU to facilitate sophisticated graphics effects and even scientific applications. In general, the GPU has become a powerful device for the execution of data-parallel, arithmetic (versus memory) intensive applications in which the same operations are carried out on many elements of data in parallel. Example applications include the iterative solution of PDEs, video processing, machine learning, and 3-D medical imaging.

With the recent release of NVIDIAs Compute Unified Device Architecture (CUDA), the hardware architecture of modern GPUs can be viewed as a series of Single-Instruction-Multiple-Data (SIMD) multiprocessors, each capable of processing one set of instructions on different elements of memory in one clock cycle. As of this writing, up to 128 processors are available on a single GPU. Each of these processors has available to it four sources of local memory within its multiprocessor unit:

A set of 32-bit registers for each processor Shared memory available to all processors Shared constant memory available to all processors Read-only texture cache available to all processors

Beyond the scope of each multiprocessor, read write access is available to the local video memory present on the GPU. Texture memory is considered a special case of device memory and is accessed by each processor via a texture unit which applies any special filtering or addressing requested depending on memory data format. Typically, the GPU is accessed for computation or graphics rendering through its device driver via either a graphics API (such as OpenGL or DirectX) or through specialized APIs such as CUDA (which shares the same name with nVidias hardware architecture). Since, we implemented the AHT update step using the OpenGL based computation model using Cg shaders we only provide the details of this model here.

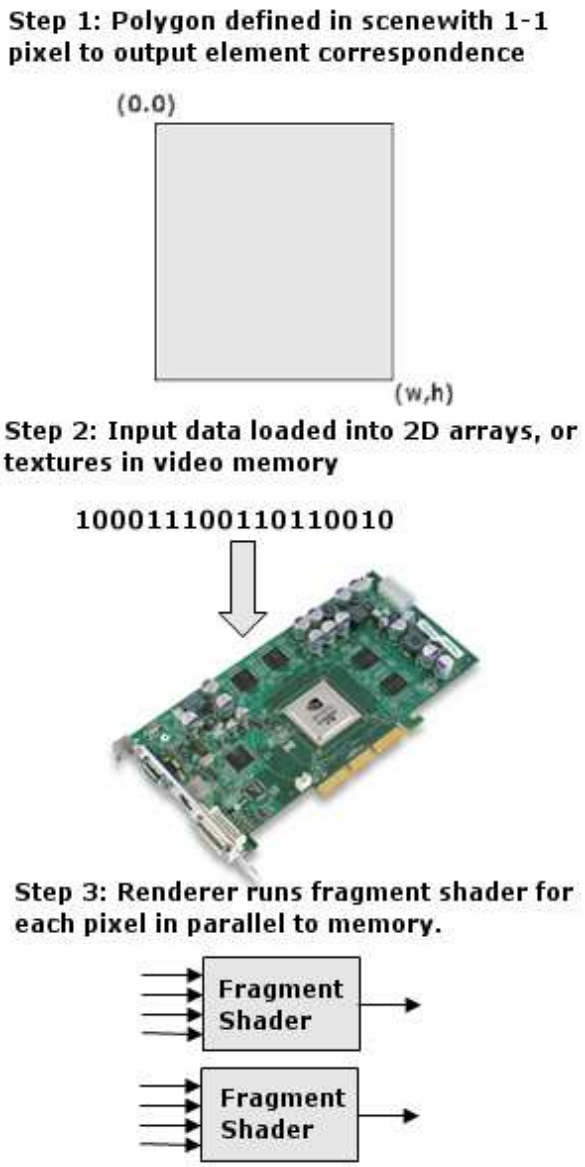


Figure 6: GPGPU Programming Model: The three steps that are required to perform general purpose computations on a GPU using Cg shaders.

The General purpose GPU (GPGPU) computing based on Cg shaders can be described in the following steps and the procedure is also outlined in Figure 6.

1. Set up the GPU to render a single polygon that fills the rendering screen such that each pixel of the polygon occupies one pixel of screen.

2. Load data (i.e. 2D arrays) to be processed onto the video card as texture.

3. Associate a *Cg fragment shader* with the polygon. Render scene to memory.

The result is that a chosen operation is performed on each element of input data and output as a result in parallel.

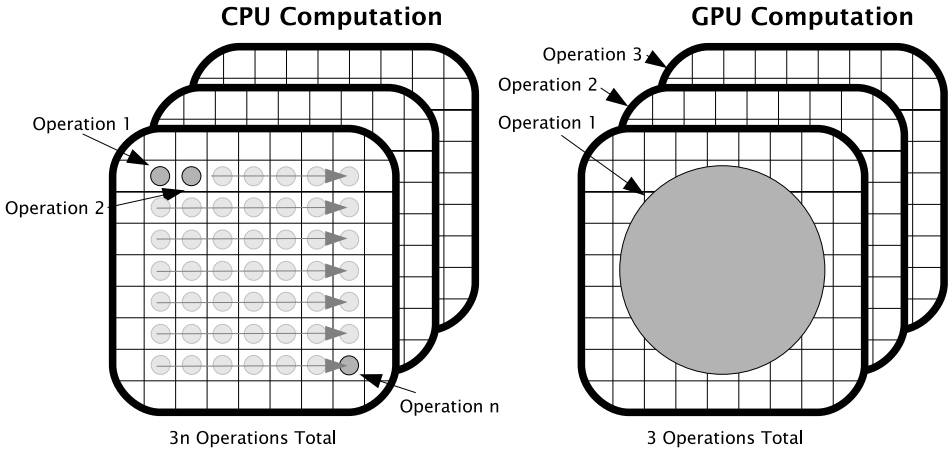


Figure 7: CPU versus GPU solution of PDEs: While the CPU computes updates on data grids one element at a time, the GPU is capable of updating entire grids in one pass due to their massively parallel architecture.

A library of 24 Cg fragment shaders was developed for simple matrix computations such as addition, subtraction, scalar multiplication etc. as well as kernel based computation of derivatives, divergence and curl. The Multigrid constituent functions of Restriction, Interpolation and Relaxation were also implemented as Cg shaders.

3.4.2 AHT Update on the GPU

Taking advantage of the benefits a parallel approach has to offer our problem, we implemented our OMT multigrid algorithm on the GPU. The GPU's advantage over the CPU in this sense is that while the CPU can execute only one or two threads of computation at a time, the GPU can execute over two orders of magnitude more. Thus, instead of sequentially computing updates on data grids one element at a time, the GPU computes updates on entire grids on each render pass, significantly improving performance (Figure 7).

The AHT algorithm is implemented on the GPU as a series of kernel operations: arithmetic computations performed component-wise over large grids of data. An

example of such an operation is the restriction operator utilized in the multigrid algorithm to down-sample data; each element of an input data grid is convolved and re-sampled to a lower resolution grid. The data flow and sequence of kernel applications involved in the OMT solver are given in Figure 8. All kernels are written in Cg in conjunction with the OpenGL/fragment shader paradigm for GPU computing as described in [77].

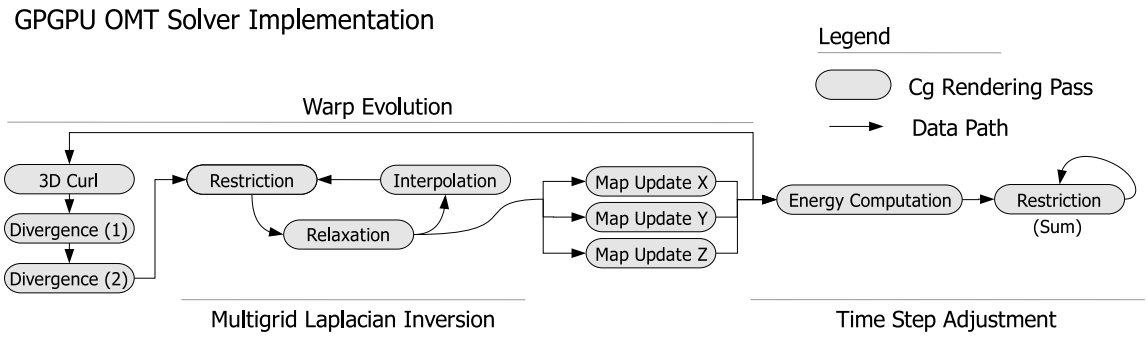


Figure 8: Outline of processing for the OMT solver conducted on the GPU. Processing occurs in two major phases: evolution of the map from source to target volumes and time step adjustment. Each gray rectangle represents one Cg kernel executed on the GPU. Arrows indicate the flow of data volumes through the Cg kernels. The entire process in the figure, above is repeated left to right until convergence.

We used the synthetic 3D test case introduced in the previous chapter to measure the performance improvements. The results are shown in Figure 9 below. On a modest Dual Xeon 1.6Ghz machine with an nVidia GeForce 8800 GX GPU (3DMark score of 7200), improvements in speed over our CPU OMT implementation reached 4826 percent on a 128^3 volume data where it converges in just 15 minutes. The GPUs available at the time of writing of this report only allow single precision computations, however, this did not affect the stability of the OMT algorithm.

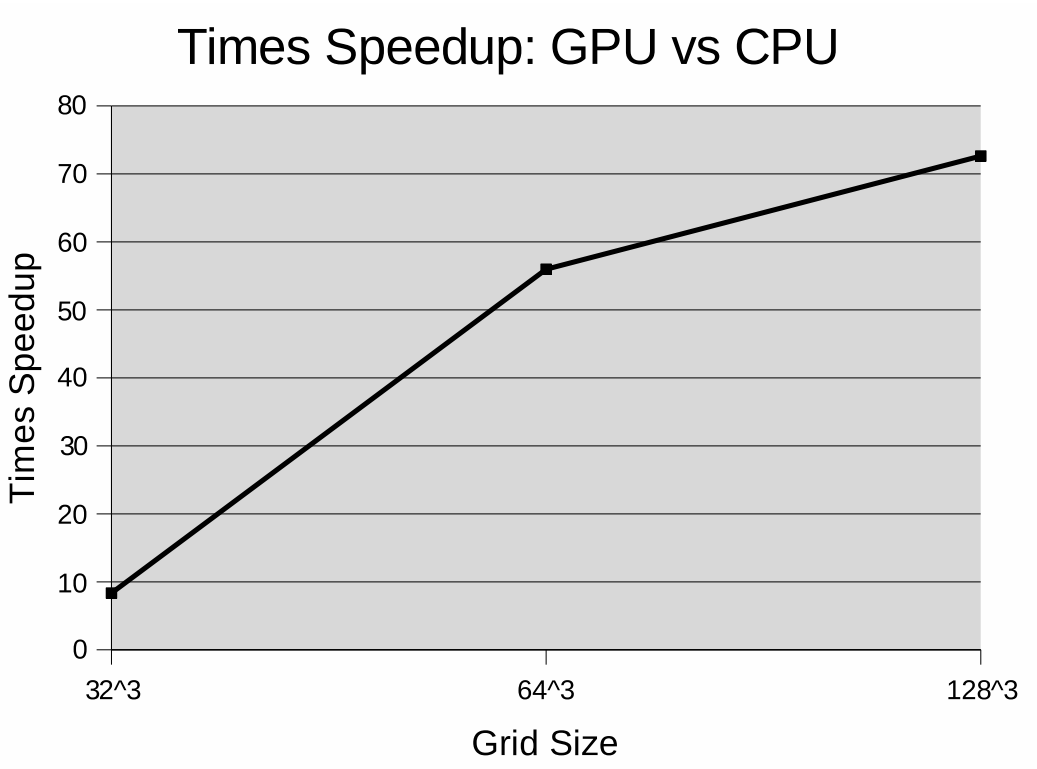


Figure 9: The GPU realizes an increasing advantage in solving the OMT problem over the CPU as grid size increases up to 128^3 sized grids.

CHAPTER IV

ADAPTIVE MESH REFINEMENT

4.1 Introduction

3D image registration is a computationally extensive problem which is commonly solved in medical imaging. The complexity of the problem stems from its size and non-linearity. In this chapter we present an approach that drastically reduces the problem size by using adaptive mesh refinement. Our approach requires special and careful discretization of the variational form on adaptive octree grids. It further requires an appropriate refinement criteria. We show that this approach can reduce the computational time by orders of magnitude compared to the non-adaptive approach.

One of the major challenges in image registration is overcoming the computational complexity of the problem. Fast and efficient numerical schemes are crucial. This is especially the case for 3D images where tens or hundred of millions of unknowns need to be evaluated. In this chapter, we address this challenge.

Several approaches towards fast implementations have been discussed in the literature: iterative solvers [23, 70], specialized direct solvers [34], fast filter techniques [17], multigrid [51, 52, 25, 57, 44]. All these techniques are combined with a multilevel strategy. However, they all use the original image grid as a finest grid. Already for moderate sized 3D images this results in large degrees of freedom. For an example, for 128^3 images, one already ends up with roughly 6 million unknowns. Thus, even a super fast implementation of a multilevel/multigrid method might be too slow in clinical application. In this chapter we propose a strategy to reduce the size of the problem by using adaptive multilevel mesh refinement. The idea is hardly new for numerical methods for partial differential equations (PDEs); see [83, 30, 7] and

reference therein. Nevertheless, the use of adaptive meshing to inverse problems is a relatively new field with very little references; see [6, 8, 9, 87]. Some relevant work on octree based image registration is in [88, 59, 43]. In [85] non-uniform octree meshes for finite-element computations were proposed with applications in solution of elliptic partial differential equations and elastic image registration. In [89, 59] the displacement field was discretized using quadtree splines and in [88] a 2D surface was embedded in 3D and represented using an octree. Other relevant contributions using octrees in image processing has been made in the field of computer graphics [63, 64]. In particular, the work of Losasso *et al.* on octree discretization demonstrates that images of fine-detail, flows, and smoke can be represented efficiently and reliably with this type of data structure.

In the following we present a multilevel adaptive mesh refinement method for non-rigid image registration using Optimal Mass Transport. We use octrees as a basic structure for the underlying displacement field and discretize the optimization problem on an octree. The goal is to represent a less complex transformation by a smaller number of unknowns. An extreme example is a translation or shift, where the complete transformation can be represented by only three unknowns. Note that the octree structure is used for the transformation, while we use the original high resolution representation for the given images. Further acceleration of the method proposed here can be obtained by using an image-pyramid structure however we choose to concentrate on the discretization of the transformation assuming a fixed image size.

4.2 Octree Data Structure and Discretization

In this section we discuss octree based discretization of the image registration problem. Following [4], we envision a uniform underlying coarse grid Ω_H with cell width H and a uniform underlying fine grid Ω_h with cell width h; see Figure 10. We assume that

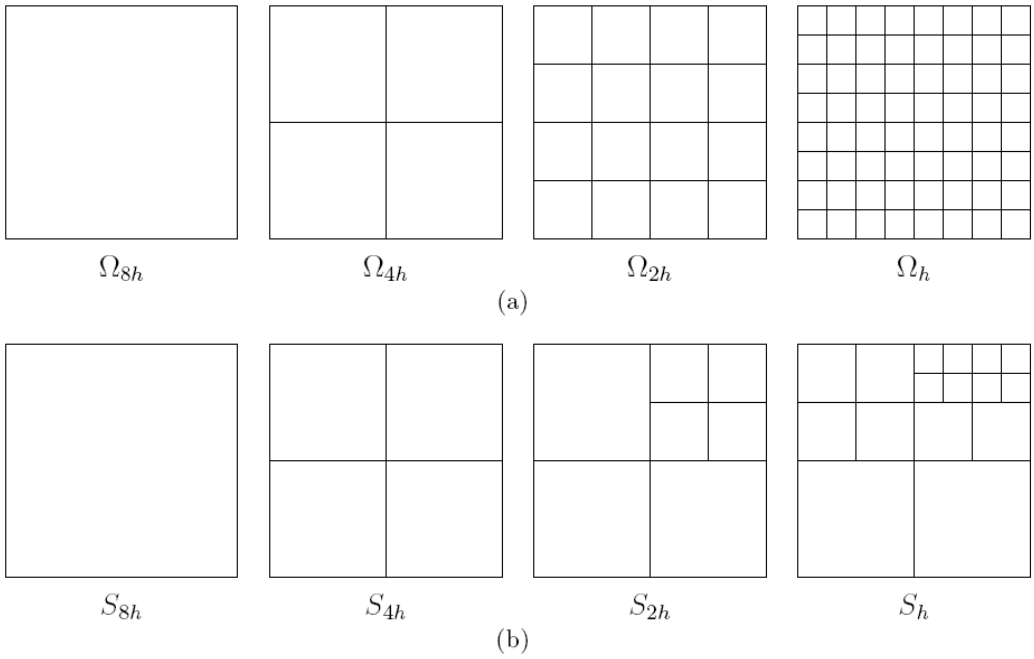


Figure 10: 2D example for grid refinement. (a) regular, refinement (b) adaptive.

$H = 2^L h$, where L denotes the total number of refinement levels. The fine grid is basically the voxel grid of the images and the coarse grid is inexpensive to work on while still producing a meaningful coarse grid solution that can serve as a starting guess for a refined level.

4.2.1 Octree Data Structure

In contrast to the regular grids, the octree grid S_h is composed of square cells of different size. Each of these cells can have a width 2_j^h where $0 \leq j \leq L$. Thus, S_h is nested in Ωh. To make the data structure easier to access, we limit the ratio of widths of adjacent cells by two. This results in a tree structure, where each node (cell) has up to eight children in 3D and four for the 2D case; see Figure 10 for an example. The grid structure is then stored as a sparse array. The size of each cell is stored in the upper left corner of the array. This allows us to quickly find neighbors, which is a major operation in the computational process. This data structure is closely related to the one suggested in [53]. For example, for the sparse grid S_{2h} presented

in Figure 10 the non-zero entries are stored as,

$$S_{2h} = \begin{array}{|c|c|c|c|c|c|c|c|}\hline 4 & & & & 2 & & 2 & \\ \hline & & & & & & & \\ \hline & & & & 2 & & 2 & \\ \hline & & & & & & & \\ \hline 4 & & & & 4 & & & \\ \hline & & & & & & & \\ \hline & & & & & & & \\ \hline & & & & & & & \\ \hline \end{array}$$

4.2.2 Discretization

Given a particular octree grid one has to decide on where to discretize the different variables. In this work, we have chosen a nodal grid based discretization which implies that all variables are discretized at the nodes. This discretization is substantially simpler to work with and implement, and second order accuracy can be easily obtained even on octrees. For ease of presentation, we derive our discretizations in 2D, the 3D extension is straightforward. As we have seen in our discretization scheme on regular grids previously, in order to discretize the objective function on an octree grid S^h, two quantities need to be computed properly. First, we need to approximate $\int \mu_0 |u|^2 \, dx$ and second the approximation of $\int |\nabla \times u|^2 \, dx$ is needed.

4.2.2.1 Transferring μ_0 to the Octree

Based on our assumption that μ_0 exists on the cell centers, we need to make sure it is interpolated to the nodes of the octree grid S^h. If A_c^n is defined as the averaging matrix from cell-centers to nodes and Q_S^Ω is the operator that maps a function from octree grid to the image grid then,

$$\mu_0^S = A_c^n \left[v \odot \left(Q_S^\Omega \mu_0 \right) \right] \tag{4.2.1}$$

where, v is the vector collecting all cell volumes and $\odot$ is the element-wise product.

4.2.2.2 *Discretizing the $\nabla\times$ Operator*

We focus on an arbitrary component u_l of the displacement. Consider the quadtree (2D octree) cell depicted in Figure 11 with cell-center "•" at position (x_1, x_2) and cell-width $2h$. In the nodal discretization all the components of u are discretize on the nodes. The partial derivatives are thus naturally discretized to second order accuracy along the centers of the edges of each cell, i.e.,

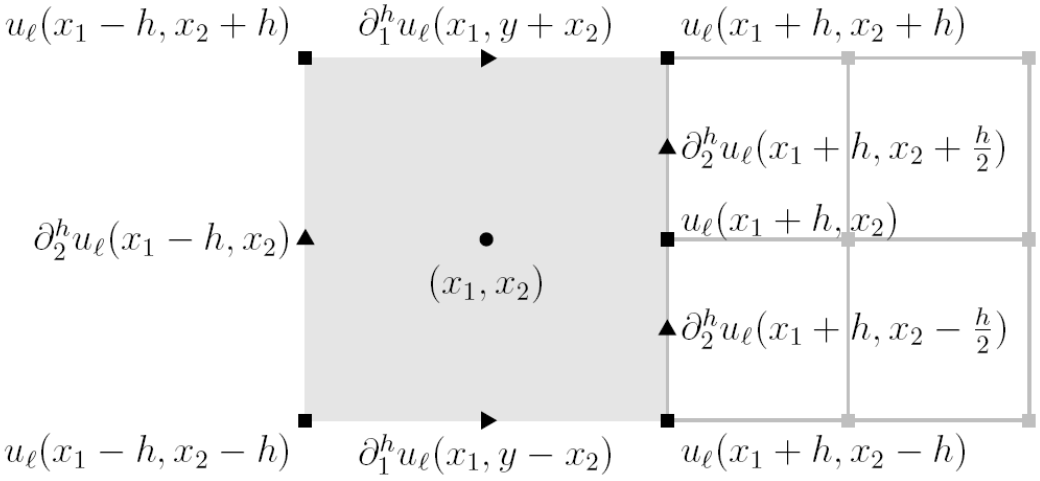


Figure 11: Discretization of ∇u_l

$$\partial_1 u_l(x_1, x_2) = \partial_1^h u_l(x_1, x_2) + \mathcal{O}(h^2) \text{ and } \partial_2 u_l(x_1, x_2) = \partial_2^h u_l(x_1, x_2) + \mathcal{O}(h^2),$$

with ∂_j^h the standard central finite differences approximation and $l = 1, 2$. Thus, for the quadtree in Figure 11 we obtain the second order approximations

$$\begin{aligned}
\partial_1^h u_l(x_1, x_2+h) &= \frac{u_l(x_1+h, x_2+h) - u_l(x_1-h, x_2+h)}{2h}, \\
\partial_2^h u_l(x_1, x_2-h) &= \frac{u_l(x_1+h, x_2-h) - u_l(x_1-h, x_2-h)}{2h}, \\
\partial_2^h u_l(x_1-h, x_2) &= \frac{u_l(x_1-h, x_2+h) - u_l(x_1-h, x_2-h)}{2h}, \\
\partial_2^h u_l(x_1+h, x_2-h/2) &= \frac{u_l(x_1-h, x_2) - u_l(x_1-h, x_2-h)}{h}, \\
\partial_2^h u_l(x_1+h, x_2+h/2) &= \frac{u_l(x_1-h, x_2+h) - u_l(x_1-h, x_2)}{h},
\end{aligned}$$

Using this second order difference scheme, we can discretize the gradient of u_l on the quadtree edges. For ease of presentation, we derive a matrix representation for the discrete $\nabla\times$ operator. Let, $\nabla^h = \left[D_1^h, D_2^h\right]$ with $D_j^h u_l^h$ the collection of $\partial_j^h u_l\left(x_1, x_2\right)$ for all discretization points, and A_e^c be an average matrix from edges to cell-center of each cell, we can write C, the **curl** matrix as,

$$C = A_e^c \begin{bmatrix} D_2^h & -D_1^h \end{bmatrix} \tag{4.2.2}$$

which for 3D is given by,

$$C = A_e^c \begin{bmatrix} 0 & D_3^h & -D_1^h \\ -D_3^h & 0 & D_1^h \\ D_2^h & -D_1^h & 0 \end{bmatrix} \tag{4.2.3}$$

4.3 Computing the Projection to Mass Preserving (MP) Constraint

In this section we present the modified algorithm to compute the projection to the MP constraint as described in last chapter. It can easily be shown that the Lagrangian to compute the minimization problem on the Octree grid can be written as following:

$$L\left(u, p\right) = \frac{\alpha}{2} u^T M_{\mu_0 S} u + \frac{\beta}{2} u^T C^T V C u + p^T c\left(u\right) \tag{4.3.1}$$

where, $c\left(u\right) = 0$ is the mass preservation equality constraint and p is the vector of Lagrange multipliers. $M_{\mu_0 S}$ is mass matrix given by $I \otimes \mu_0^S$ where $\otimes$ is the Kronecker product and I is an identity matrix of size $N \times N$ (where, $N = 3$ for 3D problems). Similarly, V is the volume matrix given by $I \otimes v$ and C is the **curl** matrix as defined above.

In order to solve this system we use the inexact SQP formulation given as:

$$\begin{bmatrix} H & c_u^T \\ c_u & -S \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} \partial L/\partial u \\ \partial L/\partial p \end{bmatrix} \tag{4.3.2}$$

where,

H is approximated by $\alpha M_{\mu_{0S}} + \beta C^T V C$

S is a regularization matrix

c_u is the Jacobian of the constraint equation.

also,

$\partial L/\partial u = \alpha M_{\mu_{0S}} u + \beta C^T V C u + c_u^T p$

$\partial L/\partial p = c\,(u)$

Re-writing this system in standard KKT form we get:

$$\begin{bmatrix} A & B^T \\ B & -S \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \tag{4.3.3}$$

Here we propose to use the augmented Lagrangian algorithm as it breaks down the problem into smaller unconstrained problems. Additionally, this method is less sensitive to the nullity of A and can also work well if there is near singularity [37]. Multiplying the second block-row of the system above by γB (where γ is a positive scalar), and adding the resulting equation to the first block equation of the system, we obtain

$$\begin{bmatrix} A + \gamma B^T B & B^T \\ B & -S \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} r_1 + \gamma B^T r_2 \\ r_2 \end{bmatrix} \tag{4.3.4}$$

A decomposition of KKT system may be defined as following:

$$\begin{bmatrix} A_{aug} & B^T \\ B & 0 \end{bmatrix} = \begin{bmatrix} A_{aug} & 0 \\ B & -\mu I \end{bmatrix} - \begin{bmatrix} 0 & -B^T \\ 0 & -\mu I \end{bmatrix} = M(\gamma) - N(\gamma)$$

For a fixed γ the convergence properties are given by the iteration matrix

$$G = M(\gamma)^{-1} N(\gamma)$$

A simple calculation shows that

$$G = \begin{pmatrix} 0 & (A + \gamma B^\top B)^{-1} B^\top \\ 0 & I - \gamma^{-1} B (A + \gamma B^\top B)^{-1} B^\top \end{pmatrix}$$

The eigenvalues of G are determined by the eigenvalues of the Schur complement

$$S(\gamma) = B^\top (A + \gamma B^\top B)^{-1} B^\top$$

The properties of this Schur complement are studied in detail in [37]. It is clear that for a sufficiently large γ the iteration converges linearly. Nevertheless, it should be also clear that if γ is too large, the convergence can be painfully small.

Now using the splitting of KKT system defined above we can write an iterative solver using the following methodology:

$$\begin{bmatrix} A_{aug} & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} r_1' \\ r_2 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} A_{aug} & 0 \\ B & -\mu I \end{bmatrix} \begin{bmatrix} u_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & -B^T \\ 0 & -\mu I \end{bmatrix} \begin{bmatrix} u_k \\ p_k \end{bmatrix} + \begin{bmatrix} r_1' \\ r_2 \end{bmatrix}$$

Which results in the following iterative scheme:

$$A_{aug} u^{k+1} = -B^T p^k + r_1' \tag{4.3.5}$$

$$p^{k+1} = p^k - \mu^{-1} * \left(r_2 - B u^{k+1}\right) \tag{4.3.6}$$

This iterative scheme can now be employed in finding the projection. The projection algorithm is outlined below in Algorithm 3:

Algorithm 3 Projection to MP constraint:

Compute $A = \alpha M_{\mu_{0S}} + \beta C^T V C + \gamma c_u^T c_u$
Initialize $p^k = 0$
for $ii = 1$ to $maxiter$ **do**
  Compute MP constraint $c^k = \left|\nabla u^k\right| \mu_1\left(u^k\right) - \mu_0$
  Compute r_1 where, $r_1 = \alpha M_{\mu_{0S}} u^k + \beta C^T V C u + c_u^T p^k + \gamma c_u^T c^k$
  Compute r_2 where, $r_2 = c^k$
  Set $B = c_u$
  Set $dp^k = 0$
  Solve KKT System $[du, dp] = kktSol\left(A, B, dp^k, r_1, r_2\right)$
  $\rightarrow du = A^{-1}\left(r_1 - B^T dp^k\right)$ – Eq. (4.3.5)
  $\rightarrow dp^{k+1} = dp^k - 1/\mu * (r_2 - Bdu)$ – Eq. (4.3.6)
  if $\|du\|_\infty > 1$ **then**
    $du = du/\|du\|_\infty$
  end if
  line search: set $\eta = 1$
  while true **do**
    $u^{k+1} = u^k - \eta du$
    $p^{k+1} = p^k - \eta dp^{k+1}$
    Compute constraint. $c^{k+1} = \left|\nabla u^{k+1}\right| \mu_1\left(u^{k+1}\right) - \mu_0$
    if $\left(\|c^{k+1}\| < \|c^k\|\right) AND \left(min\left|\nabla u^{k+1}\right| > 0\right)$ **then**
      Break
    end if
    $\eta \Longleftarrow \eta/2$
  end while
  Update $\gamma = 1.5\gamma$
end for

4.4 Adaptive Mesh Refinement

The cost of the optimization process is directly impacted by the size of the problem and the initial guess for the solution. Adaptive multilevel refinement methods are targeted to achieve a low-cost good starting guess by using coarse grids, and to reduce the size of the discrete fine grid problem by using adaptive nested grids that refine only in areas where the error in the solution is large. Unfortunately, finding a unique refinement criterion that works for different problems is rather difficult; see,

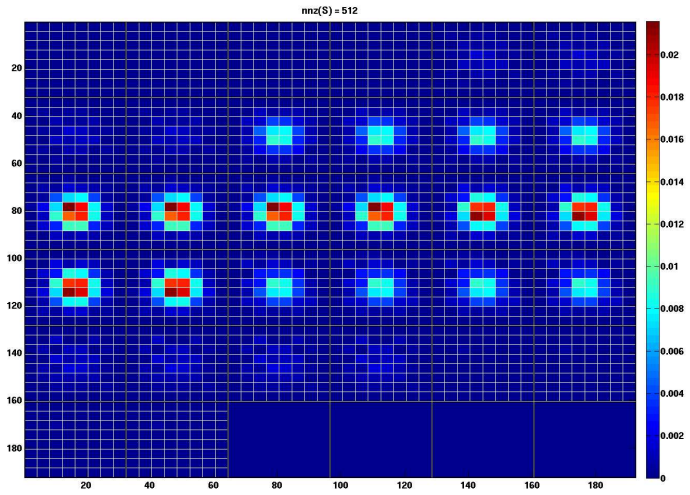
Table 3: Speed-up results for the 3D synthetic example

Grid size	Without OcTree (MINRES) Time(min)	With OcTree (Matrix Splitting) Time(min)	Speed-up
2^{-4}	1.8	0.28	6.4
2^{-5}	32.4	3.2	10.1
2^{-6}	712.3	14	50.8
2^{-7}	13816	36	383.7

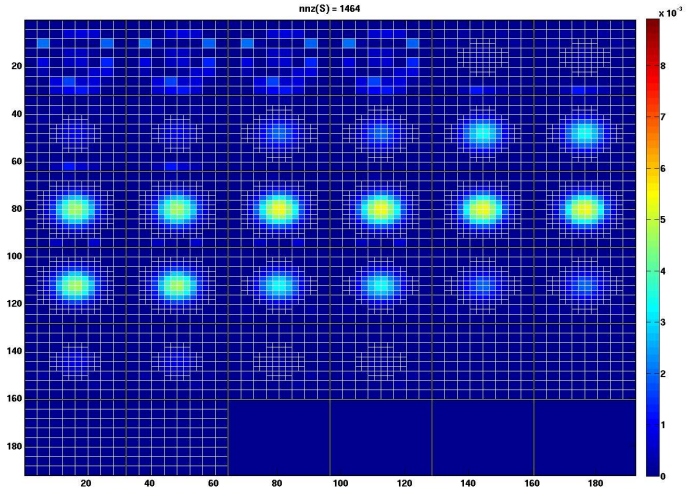
e.g., [93]. We base our refinement criterion on the value of the MP constraint $\left|c\left(u^k\right)\right|$ at any iteration k. Since, the solution of the projection algorithm is directly measured by the constraint, we want to refine in areas where it is large. In order to decide if $\left|c\left(u^k\right)\right|$ is large we use a parameter τ and refine every cell that satisfies $\left|c\left(u^k\right)\right| > \tau$. The refinement process is terminated when $\left|c\left(u^k\right)\right| < \tau$ for all cells. Figure 12 below shows the value of the refinement criterion and the corresponding octree meshes at different levels of the multi-level method.

4.5 Numerical Experiments

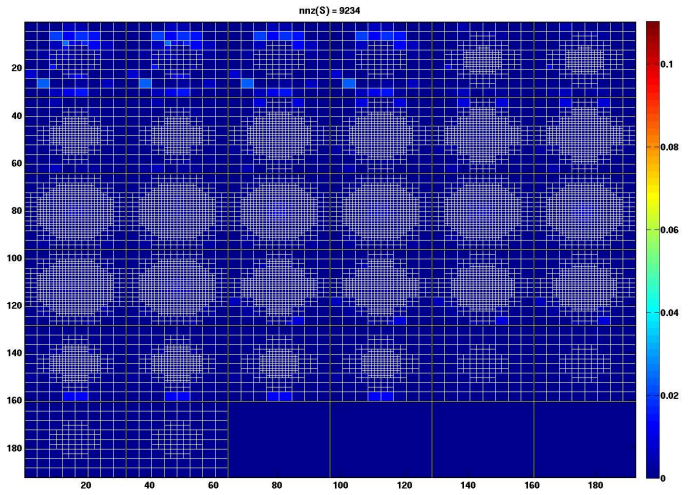
To demonstrate our method, we used the 3D synthetic data of the known deformation field example given previously. The goal of the experiments was to compare matrix-splitting based projection algorithm computed on an octree to the GMRES based projection algorithm without and octree. The general setup for each test case is as follows. We performed OMT computation with different grid sizes. The tolerance τ was chosen to be 0.10 and all experiments were started on a coarse mesh consisting of 7^3 cells resulting in 3×8^3 unknowns. The stopping criteria for the optimization on a single level was when the maximum difference of consecutive iterates was below 0.1 voxel width. The linear systems were solved to a precision of 10^5 using a preconditioned conjugate gradient method. Table 3 shows the results of these experiments. All computations were performed on on Intel Dual Xeon 1.6GHz machine.



(a) OcTree Level 1



(b) OcTree Level 2



(c) OcTree Level 3

Figure 12: MP Constraint Minimization($\left|c\left(u^k\right)\right|$) with Mesh Refinement

CHAPTER V

IMAGE MORPHING AND REGISTRATION

Our interest in OMT is motivated by its applications in image processing, specifically registration and morphing. Image registration and morphing are amongst the most common image processing problems. Registration is necessary in order to compare or integrate image data obtained from different measurements, on the other hand, image morphing is a class of techniques that deals with the metamorphosis of one image into another (also known as image interpolation). Given two related images, these two techniques can be used together to generate a sequence of intermediate images in which an image gradually changes into the other over time. Critical to the success of this process is the quality of the warping function generated by the registration procedure. Morphing is achieved by shifting all pixels in the image according to the vector field defined by this warping function. Desired results cannot be achieved if the underlying warping function is not correct.

OMT has a number of distinguishing characteristics that make it an attractive candidate for the above mentioned applications: **(1)** it is a parameter free method and no landmarks need be specified, **(2)** it is symmetrical (the mapping from image A to image B is the inverse of the mapping from B to A), **(3)** its solution is unique (no local minima), **(4)** it can register images where brightness constancy is an invalid assumption, and **(5)** OMT is specifically designed to take into account changes in densities that result from changes in area or volume.

5.1 Image Morphing

The OMT approach can also be applied to image morphing that is also a widely used technique in image processing. It essentially deals with creating a sequence of images

that smoothly and realistically interpolate between a pair of images. The term *morph* is, therefore, a short for *metamorphosis*. The whole procedure can thus be viewed as image interpolation in the time domain. All image morphing methods seek to find a reasonable warping function between a source and target image pair that can then be used generate the intermediate images in the sequence through a simple method called *cross-dissolving*.

There have been many algorithms proposed for image morphing in literature. We will just sample some of the key approaches most relevant to the present work. The most prominent amongst these fall under the categories of mesh warping [97], field morphing [10], radial basis functions [3] , thin plate splines [61], energy minimization [60] and multilevel free-form deformations [62]. A common feature present in almost all of these techniques is some sort of feature selection mechanism. This is employed to set up correspondences between the two images from which a warping function can be generated. Feature specification has been in general the most tedious aspect of morphing, and there is usually a tradeoff between the complexity of feature specification and warp generation.

The proposed approach in this work can be categorized as an intensity-based technique closely related to such ideas in the medical registration literature. Here one uses only information given by the images such as pixel intensities to perform the warping. A successful intensity-based algorithm can achieve automatic morphing without user inputs or prior assumptions on special shapes or features of the objects in the image. See [92] for a review of the literature and an extensive list of references on this subject.

5.1.1 Algorithm and Results

In this section we present some image morphing results using the proposed OMT framework. The mass transport model seems to fit a wide range of dynamic imagery

such as clouds, flames and deforming objects. Here again the warping map is obtained between the source and target images using the OMT framework. Once we have the optimal mass preserving map. The morphed intermediate images between the source and target images can be created using the warp calculated from the following relationship [15, 36]:

$$X(x,t) = (1-t) * x + t * u^*(x) \tag{5.1.1}$$

where $X(x,t)$ defines our continuous warping map between the source and target image at time t, and u^* denotes the optimal mass preserving map. Note that when $t = 0, X$ is the identity map and when $t = 1$, it is the optimal map u^*. Therefore, parameter t acts as the transition rate. An interpolated image for any $t \in [0,1]$ can then be obtained by::

$$I^t(X(x,t)) = (1-t) * I_0(x) + t * I_1(u^*(x)) \tag{5.1.2}$$

The mass preservation property can be ensured simply by shading the pixels in the interpolated images according to $|D(X^t)^{-1}|\mu_0 \circ (X^t)^{-1}$.

We applied the aforementioned procedure to two different cases. In the first case we registered two images taken from a sequence of ink diffusion in liquid. In Figure (13), we show the results for the ink diffusion image pair (512×512 pixels). The top left and bottom right in the smaller images are the source ($t = 0$) and target ($t = 1$) images, respectively and all the in-between images are interpolated using Equation (5.1.2). The image at the bottom is a quiver plot of the warp superimposed on target image.The second example shown in Figure (14) is of two images extracted from a SOHO solar proton shower movie sequence. The layout of this figure is same as in Figure (13).

In Figure (15) we show that classical non-rigid registration based on optical flow fails to warp the flare images. The procedure cannot account for the new information present in the second image versus the first image.

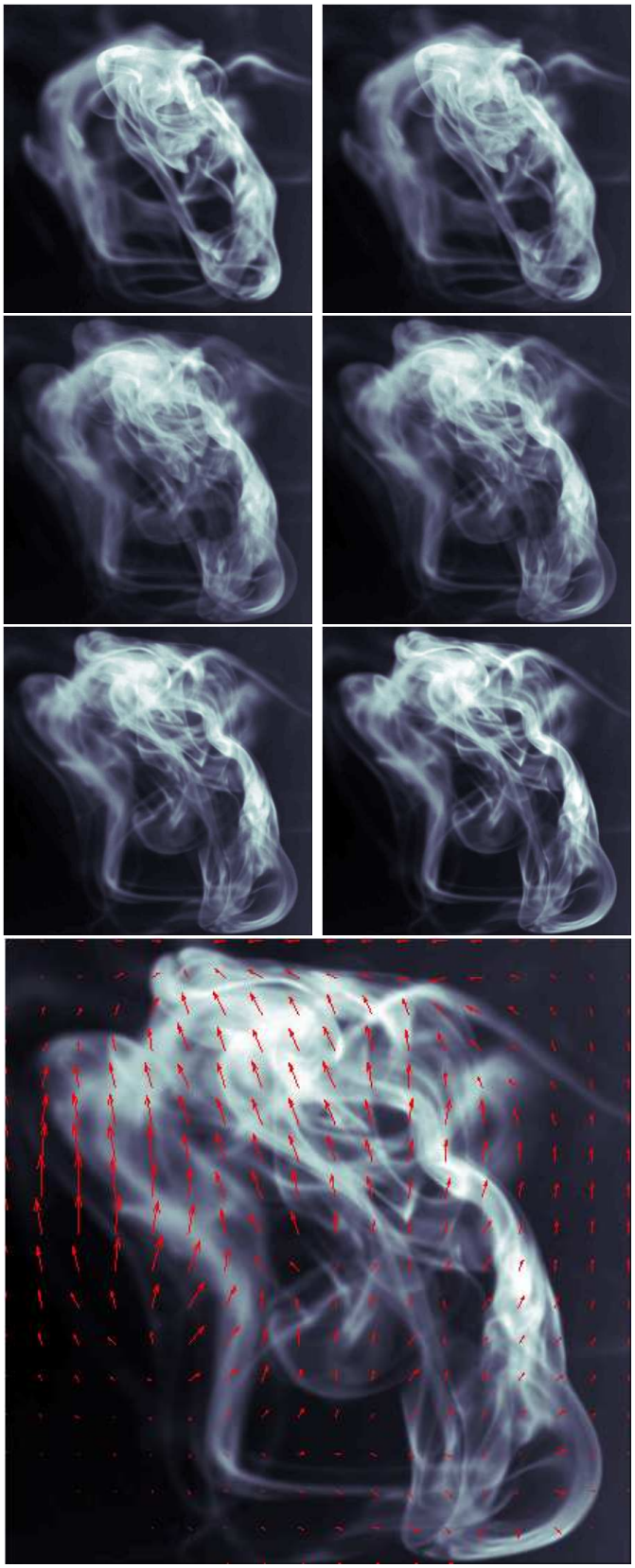


Figure 13: 2D Morphing Example 1. Morphing results for ink diffusion image pair. The images at the top show individual frames from the morph and the image at the bottom shows the deformation vector field overlaid on the source image.

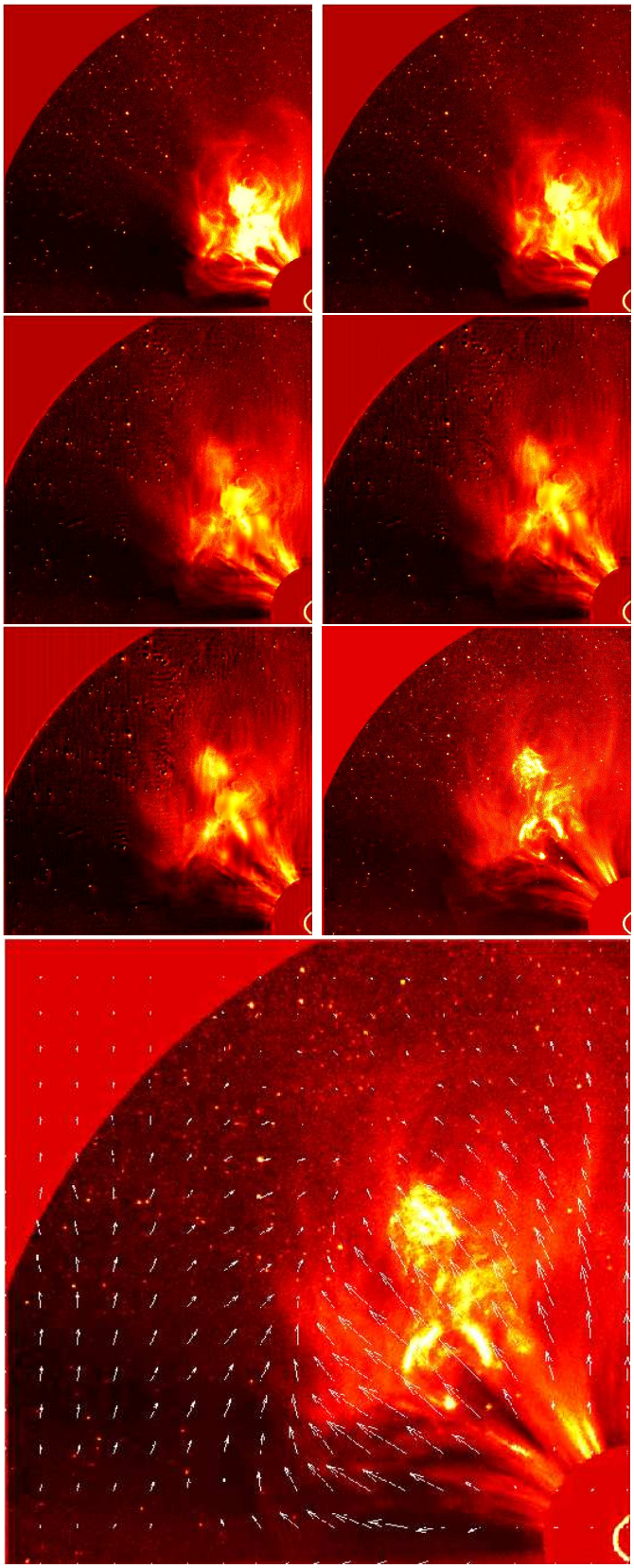


Figure 14: 2D Morphing Example 2. Morphing results for solar flare image pair. The images at the top show individual frames from the morph and the image at the bottom shows the deformation vector field overlaid on the source image.

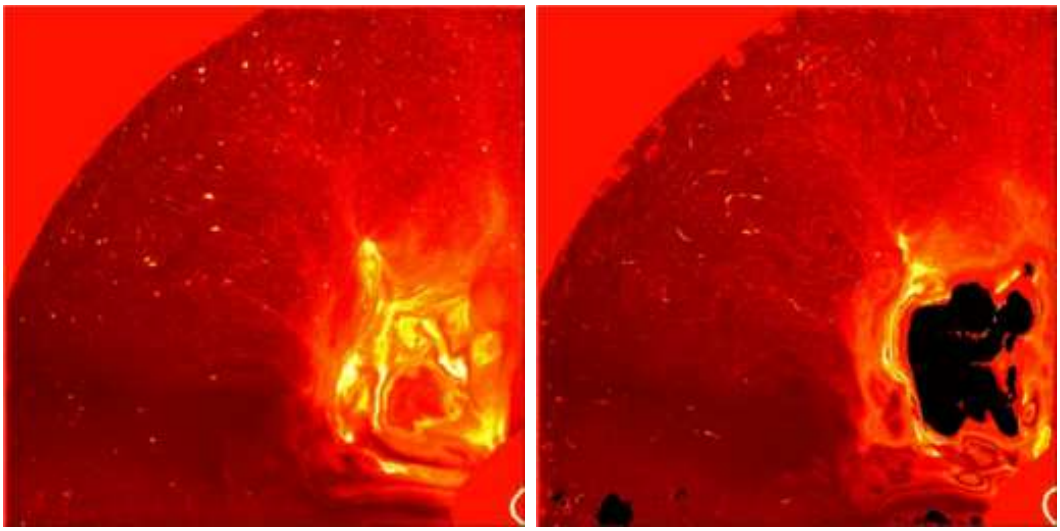


Figure 15: Optical Flow Fails. Classical non-rigid registration based on calculation of optical flow fails to morph the solar flare images.

5.2 Image Registration

In this section, we propose a method for image warping and elastic registration based on the classical problem of optimal mass transport. Registration is the process of establishing a common geometric reference frame between two or more image data sets. In the context of medical imaging, registration allows for the incorporation of preoperative image information in a surgical setting to improve image-guided surgery and therapy. It also aids in diagnosis by allowing the concurrent use of information from multiple data sets, possibly taken at different times using different modalities and patient positions. Registration processes proceed in several steps. Typically, a measure of similarity between the data sets is established, so that one can quantify how close an image is from another after transformations are applied. Such a measure may include the similarity between pixel intensity values, as well as the proximity of predefined image features such as implanted fiducials, anatomical landmarks, surface contours, and ridge lines. Next, the transformation that maximizes the similarity between the transformed images is found. Often this transformation is given as the

solution of an optimization problem where the transformations to be considered are constrained to be of a predetermined class. Finally, once an optimal transformation is obtained, it is used to fuse the image data sets. In the context of medical imaging, this is an essential technique for improving preoperative and intraoperative information for diagnosis and image-guided therapy.

A vast amount of literature exists on image registration techniques and we refer the reader to [65, 19, 38, 46] for an overview of this field. The solution approaches to the registration problem range from statistical to computational fluid dynamics to various types of warping methodologies. Our method is in the class of warping strategies based on continuum and fluid mechanics, in which one tries to use properties of elastic materials to determine the deformation. One defines a (typically quadratic) cost functional that penalizes the mismatch between the deforming template and target (Christensen et al., 1993, 1996; Miller et al.; Bro- Nielsen and Gramkow, 1996; Thirion, 1995). In this sense, our method is closest to the registration philosophy of these works. In fact, the optimal warping map of the L2 Monge-Kantorovich may be regarded as the velocity vector field which minimizes a standard energy integral subject to the Euler continuity (mass preservation) equation (Benamou and Brenier, 2000).

Broadly speaking, image registration techniques can be classified as either "rigid" or "non-rigid". Rigid registration is usually performed when the images are assumed to be of objects that simply need to be rotated and translated with respect to one another to achieve correspondence. Non-rigid registration on the other hand is used when either through biological differences or image acquisition or both, correspondence between structures in two images cannot be achieved without some localized stretching of the images [26]. In contrast to rigid registration techniques, non-rigid registration techniques are still the subject of significant ongoing research. We approach the task of non-rigid registration by treating it as an optimal mass transport

problem. As with other non-rigid registration techniques, the computational burden associated with this problem is high, however, the use of multi-resolution approaches presented in earlier chapters help to alleviate this problem.

5.2.1 3D Non-rigid Registration

In context of image registration applications the input data to our algorithm is the source and target volumes that need to be registered. For the examples presented in this section we model the mass density for a voxel as the image intensity. However, it can also be alternatively defined as any scalar field that is related to the underlying physical model. This property can be exploited for non-rigid registration of multi-modality data as well, where sufficient anatomical correspondence exists between the source and target datasets. In order for the notion of mass preservation to be valid it is necessary that both volumes have same total mass. This is ensured by normalizing the image intensities by the respective sum of all intensity values in each volume. The normalized data is then scaled by a common factor to avoid numerical instability due to very small values. Another step in the pre-processing of input data is the addition of a small mass in the background regions where there the intensity values are zero in order to avoid a divide-by-zero while solving Equation (2.2.5). Another step necessary in context of Brain MRI registration is dealing with the inherent anisotropic nature of the data. We pre-process all brain MRI data by interpolating and re-sampling to isotropic voxels.

In the first example, we register a synthetically generated 3D sphere (128x128x128) to a deformed (dented) counterpart; see Figure 16. It can be clearly seen that our algorithm successfully captures the deformation in the sphere.

In the second case, we registered two 3D brain MRI datasets. The first data set was pre-operative while the second data set was acquired during surgery (craniotomy

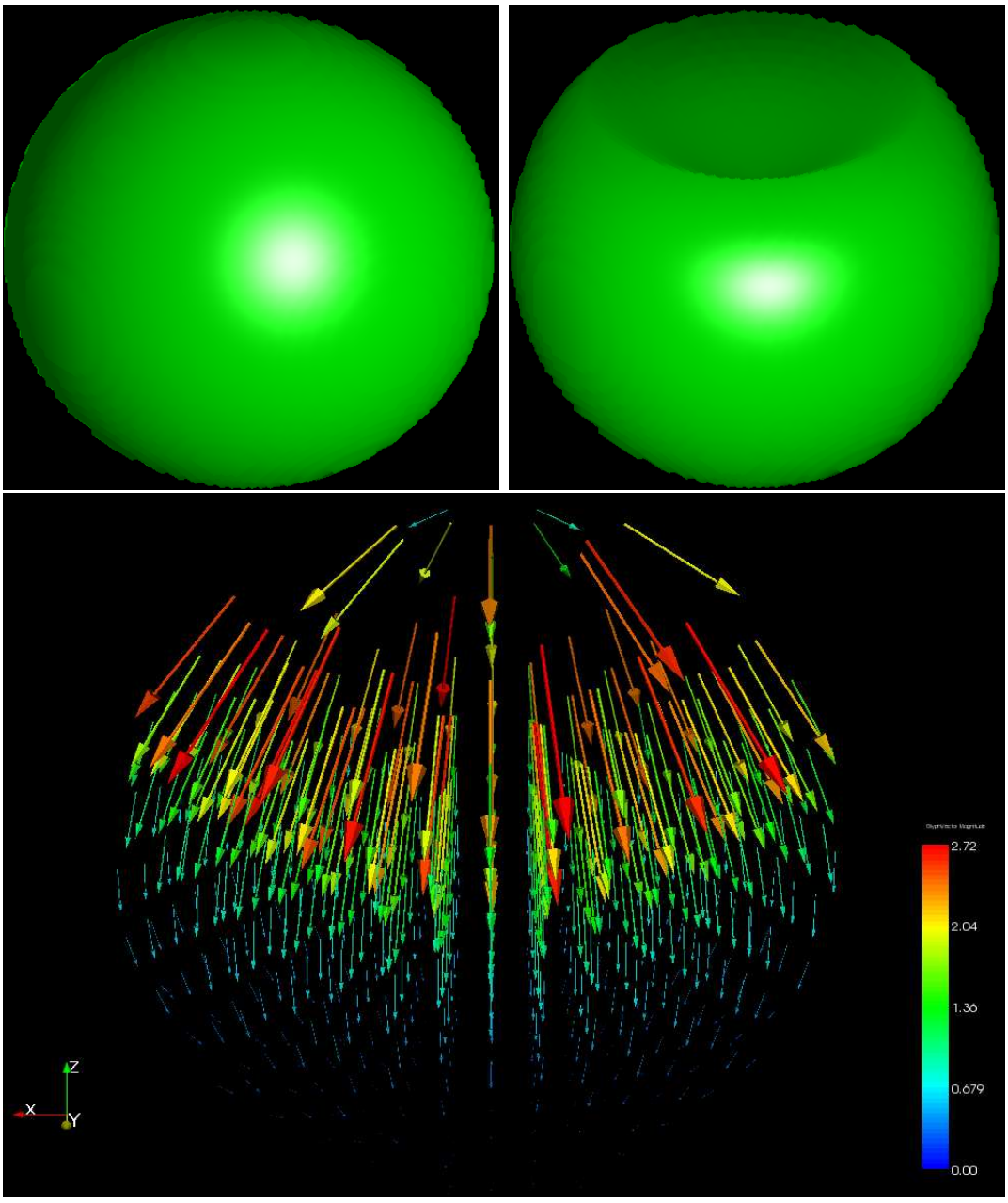


Figure 16: Synthetic Imagery Results. A sphere is mapped to its deformed counterpart. In the lower image we show the deformation vector field. It is clearly visible that the magnitude of deformation is maximum at the top where the dent is and it decays smoothly inside the sphere. Data size 128^3)

and opening of the dura). Both were resampled to 128^3 isotropic voxels and preprocessed to remove the skull. The results are shown in Figure 17

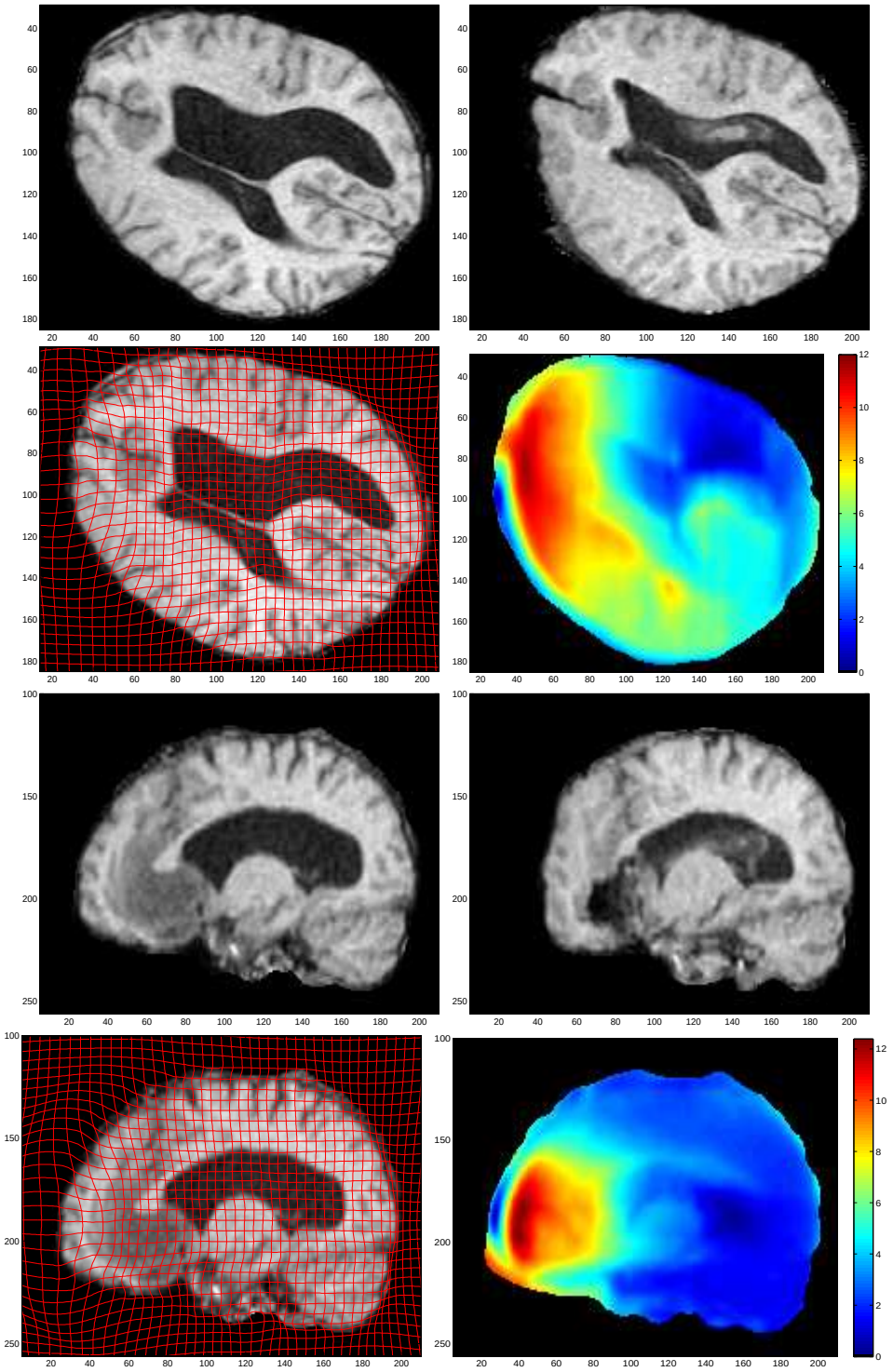


Figure 17: Brain Sag Registration. The top four figures show the registration results on an axial slice and the bottom four show results for a saggital slice from the 3D volume. The deformation due to the brain sag after carniotomy and opening of the dura is clearly visible in both the deformation grid and the magnitude of deformation plots. The gravity vector is parallel to the horizontal axis. A rigid shift can also be noticed due to slight displacement of the head during surgery.

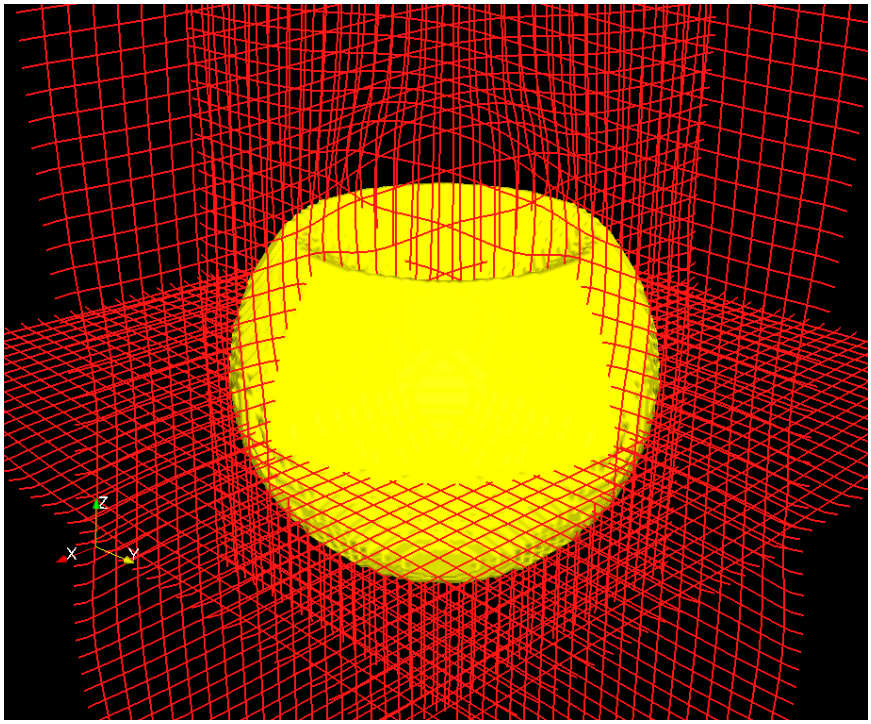


Figure 18: Sphere Registration(3D View). The deformation is visible at the dented region of the sphere. (Data size 128^3).

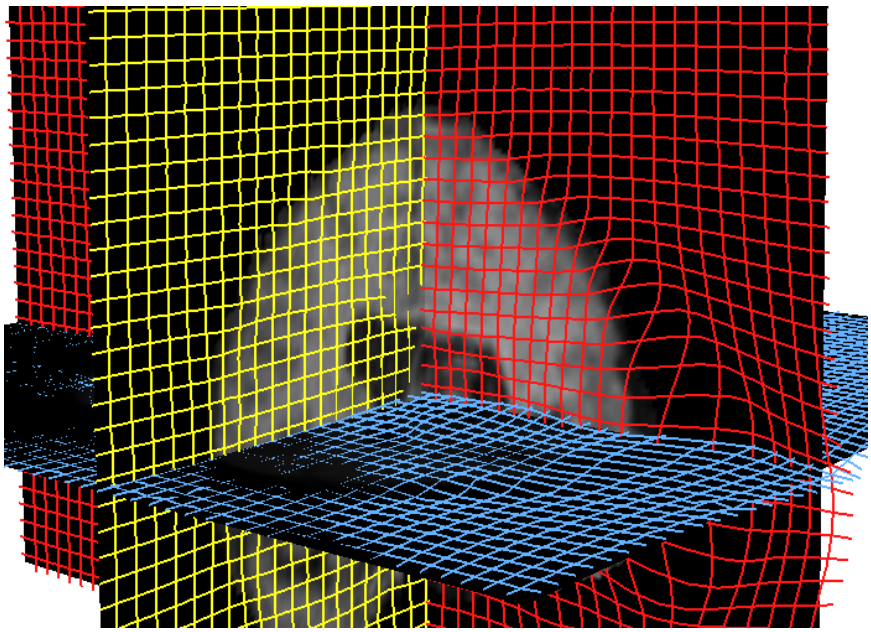


Figure 19: Brain Sag Registration(3D View). The brain sag is visible in the anterior portion of the brain. (Data size 256^3).

For each of the above examples the deformation map was computed in fewer than 20 iterations. The curl (*optimality metric*) was reduced to less than 10^{-3}, indicating convergence. This is a major improvement over the previous methods [48, 2] where thousands of iterations were required for convergence. Another advantage to our method is the explicit projection to the mass preserving constraint in each iteration that ensures that the calculated mapping always takes us from the source image to the target image.

5.2.2 Multimodal Registration of White Matter Brain Data

The elastic registration of medical scans from different acquisition sequences is becoming an important topic for many research labs that would like to continue the post-processing of medical scans acquired via the new generation of high-field-strength scanners. In this section, we introduce a parameter-free registration algorithm that is well suited for this scenario as it requires no tuning to specific acquisition sequences. The algorithm is based on the numerical scheme presented earlier in chapter 2 and 3, for computing elastic registration maps based on the minimizing flow approach to optimal mass transport. We apply the algorithm to register the white matter folds of two different scans and use the results to parcellate the cortex of the target image. To the best of our knowledge, this is the first time that the optimal mass transport function has been applied to register large 3D multimodal data sets.

5.2.2.1 Introduction

Registration is an important pre-processing step for many automatic approaches that extract cortical structures from Magnetic Resonance Images (MRI) [31, 92, 49]. Common approaches for aligning the atlas of the segmenter to the patient MRI are based on the B-spline representation [49, 82] and continuum and fluid mechanics, [24, 68, 18, 91]. The accuracy of these approaches generally depends on how well

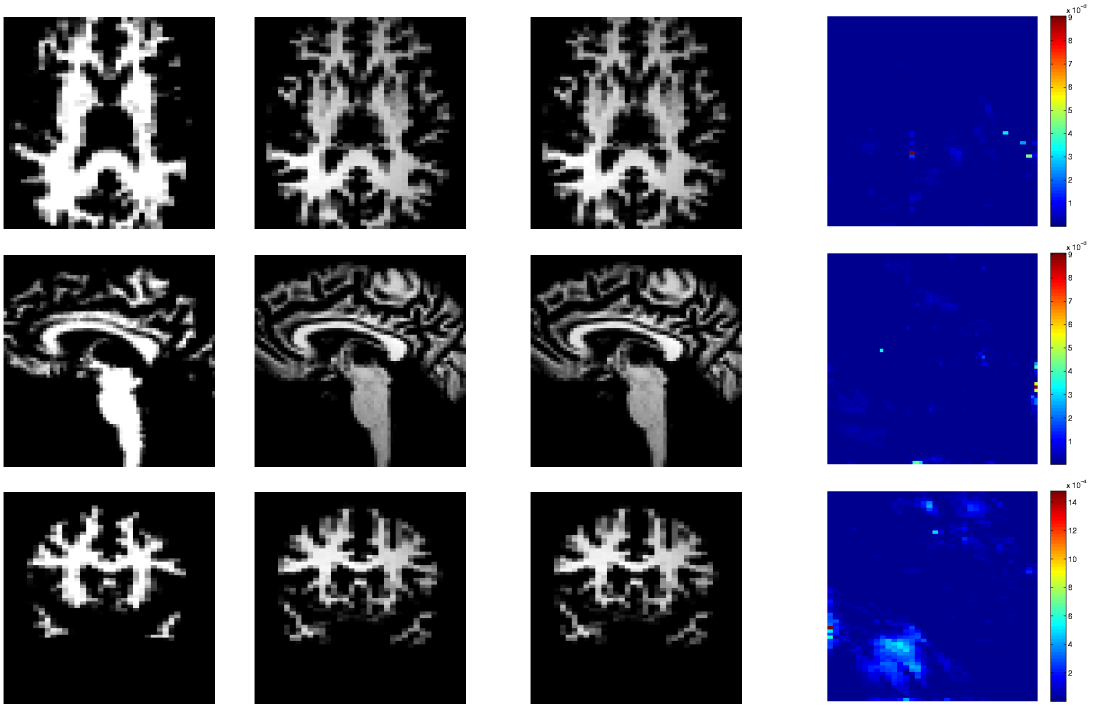
they are tuned to the sequence of patient scan. Tuning these algorithms often requires expertise about the underlying algorithm. Clinicians scanning with new acquisition sequences are therefore often concerned on how to post-process these scans. We propose a parameter-free algorithm for the registration of MRIs where we model the registration of images as an optimal mass transport problem.

5.2.2.2 Algorithm and Results

Our goal is the identification of cortical structures by mapping a publicly available atlas [32] to the scan of a patient. In our scenario, the scanning sequence of the atlas is very different from the one of the patient. The MRI of the atlas is a spoiled gradient recalled image acquired on a 1.5-Tesla General Electric Signa System (GE Medical Systems, Milwaukee) with 256x256x124 voxels and voxel dimension of 0.92 $\times$ 0.92 $\times$ 1.5 mm. The patient scan is a MPRAGE acquired on a Siemens 3T long bore machine using a 8 channel head coil. The resolution of the scan is 256 $\times$ 256 $\times$ 144 with voxel dimension 0.54 $\times$ 0.54 $\times$ 1.0 mm (See Figure 1(b).

The parcellation of the cortex can be encoded by partitioning the boundary between cortex and white matter into anatomical regions [31]. The label map of cortical structures can then be inferred from this partition by propagating the labeling along the boundary to the entire cortex. The pipeline described below will apply this concept for the parcellation of the cortex to the high resolution scan. The overall pipeline is outlined in Figure 22.

The input of the pipeline consist of the atlas, the high resolution scan as well as a segmentation of the scan into the major tissue classes. In the first step, we coarsely align the atlas to the image data using the B-spline implementation by Rohlfinger[82] with a final spacing of the grid nodes of 2.5 mm. This results in a coarse alignment of the scans. The algorithm has difficulties in mapping the folds of the white matter due to the inherent constraints of the B-spline representation. We then reduce the atlas to



(a) B-Spline (b) Patient Scan (c) Mass Transport (d) Difference ((b)-(c))

Figure 20: White Matter Registration results.

the white matter including the parcellation of the cortex along the boundary between gray and white matter (see Figure 20(a)). Afterwards, we refine the alignment of this new atlas to the white matter of the high resolution scan using our Optimal Mass Transport registration approach. Registration using Optimal Mass Transport is a highly flexible approach that is, unlike B-Splines, not constrained to a set of control points. The intensities in the two input datasets are first normalized and rescaled to make sure that both have the same total mass. The white matter registration with the proposed algorithm took just 12 iterations to converge with 2 iterations of the projection to constraint per iteration. This is a huge improvement over algorithm proposed in [48] where thousands of iterations were required for convergence with roughly the same computational complexity per iteration. The $\nabla \times u$ (convergence metric) was reduced to an order of 10^{-3} indicating an optimal map. Figure 20(c) shows the resampled images with 3D views of the corresponding deformation grid in Figure 21. The difference (Figure 20(d)) between target (Figure 20(b)) and resampled

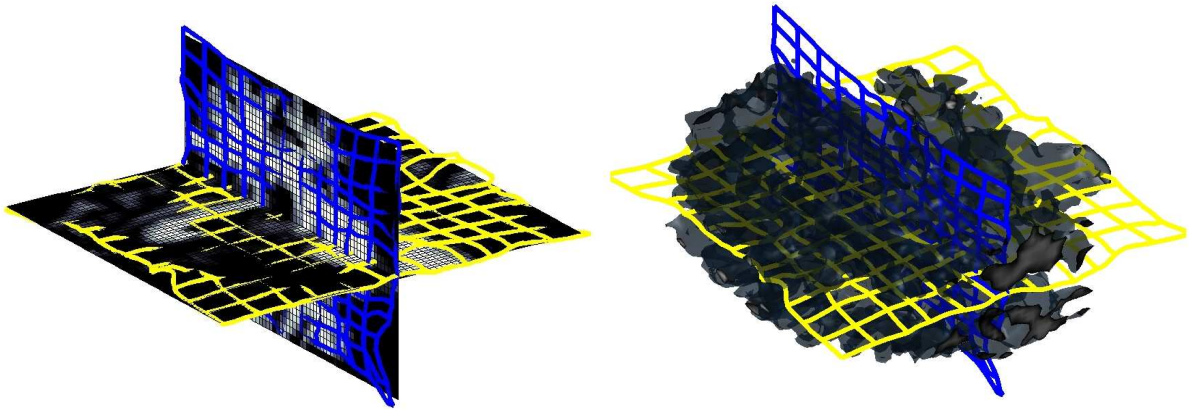


Figure 21: Deformed Grid on white matter Slices (left) and 3D volume (right).

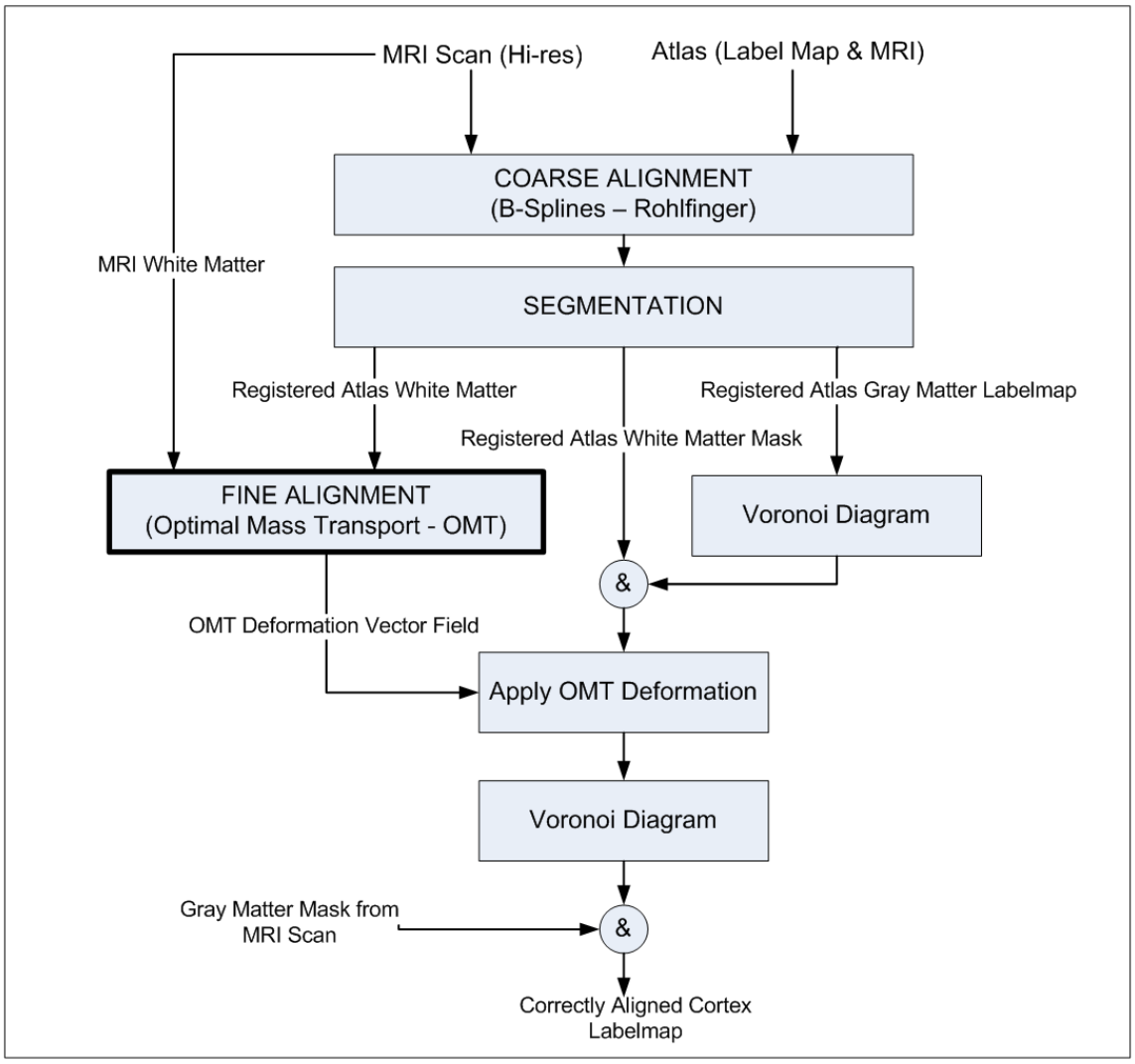
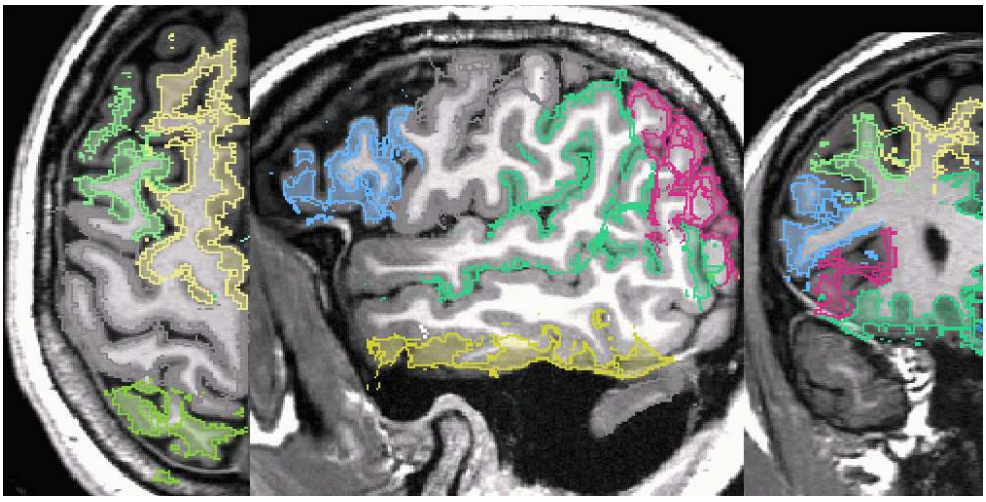


Figure 22: White Matter Registration Pipeline.

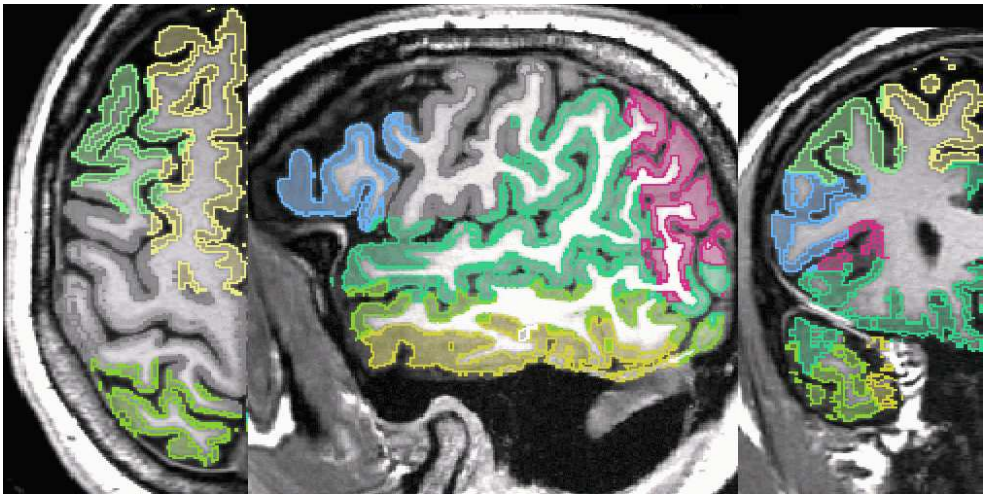
image indicates that our approach accurately aligned the folds.

After this local alignment, the folds of the atlas should perfectly align with the ones of the high resolution scan. The parcellation of the folds of the atlas, therefore, also encodes the parcellation of the same region in the high resolution scan. We then complete the cortex parcellation of the high resolution scan by confining the Voronoi diagram of the aligned atlas to the gray matter mask of the high resolution scan. The results in Figure 23 show the corresponding segmentation when applying the deformation map of the B-Spline registration and our approach to the the label map of [32], and propagating the labels to the cortex via the Voronoi diagram.

We are aware of the variety of other methods for registering and segmenting cortical structures. We also note that our segmentation results are by no means perfect. However, to the best of our knowledge, this is the first time in medical imaging that a parameter-free registration tool has been used for registering the cortical folds of 3D MRIs. The difficulties of aligning cortical folds is reflected by the large body of literature discussing this topic. Registration approaches based on continuum and fluid mechanics are often applied to this problem. However, the accuracy of these approaches generally depends on how well they are tuned to the sequence of patient scan. We view Optimal Mass Transport (OMT) as part of these types of registration approaches. Unlike the current state of the art, OMT is parameter free. It is, therefore, especially suited to align new acquisition sequences, which the other methods have not yet been tuned to.



(a) Maxwell Demons



(b) Proposed Method

Figure 23: Cortex Parecellation Results

CHAPTER VI

DYNAMIC ACTIVE CONTOUR TRACKING

6.1 Introduction

In computational vision, visual tracking remains one of the most challenging problems due to noise, clutter, occlusion, and dynamic scenes. No one technique has yet to solve this problem completely, but those that employ control-theoretic filtering techniques have proven to be quite successful. In this work, we extend one such technique by Niethammer *et al.* in which implicitly represented dynamically evolving contours are filtered using a geometric observer framework. The effectiveness of the observer hangs upon the solution of two major problems: (1) the calculation of accurate curve velocities and (2) the determination of diffeomorphic correspondence maps between curves for geometric interpolation. We propose the use of novel image registration techniques such as image warping and optimal mass transport for the solution of these problems which increase the performance of the framework and reduce algorithmic complexity. One major drawback to the original scheme, as it relies on PDE solutions, is its computational burden restricting it from real time use. We show that the framework can, in fact, run in real time by implementing our additions to the framework on the graphics processing unit (GPU) and show better execution times for these algorithms than reported in recent literature.

Visual tracking is the act of consistently locating a desired feature in each image of an input sequence. It is a critical step in numerous machine vision applications such as surveillance, driver assistance, human-computer interactions, etc. However, it is a challenging problem due to noise, clutter, occlusion, and dynamic scenes. Numerous approaches and techniques exist for the solution of the tracking problem and some of

the most successful employ control-theoretic filtering techniques.

In this chapter we introduce on one such control-theoretic tracking framework by Niethammer *et al.* [72] who proposed a deterministic observer framework for visual tracking based on non-parametric implicit (level-set) curve descriptions. This work is novel in that it provides a general framework for the filtering of implicit curves along with any amount of additional state associated with those curves without the need for parameterizations which are complex to implement in practice have their own particular drawbacks.

The above referenced work involves a continuous-discrete observer with continuous-time system dynamics and discrete-time measurements. The proposed framework is general in nature and multiple simulation models can be incorporated. Measurements can be performed through static segmentation techniques and optical flow computations.

Unfortunately, discrete-time measurements lead to the problem of geometric curve interpolation and the filtering of quantities propagated along with implicitly represented estimated curves. Interpolation and filtering are intimately linked to the correspondence problem between curves. In [72], this is done using a Laplace equation approach, establishing a one-to-one correspondence between measured and estimated curves to determine unique "distances" between points and to exchange information between them for dynamic filtering. The distance measurements allow for a geometric interpolation between measured and estimated curves, facilitating geometric, intuitively tunable gains for position filtering.

In our work, we use the Monge-Kantorovich formulation of the optimal mass transport (OMT) problem for the purpose of establishing the diffeomorphic correspondence maps between the estimated and measured curves instead of the Laplace equation approach which requires complex domain decompositions in order compute the correct map. The OMT method, on the other hand, is an elegant solution to

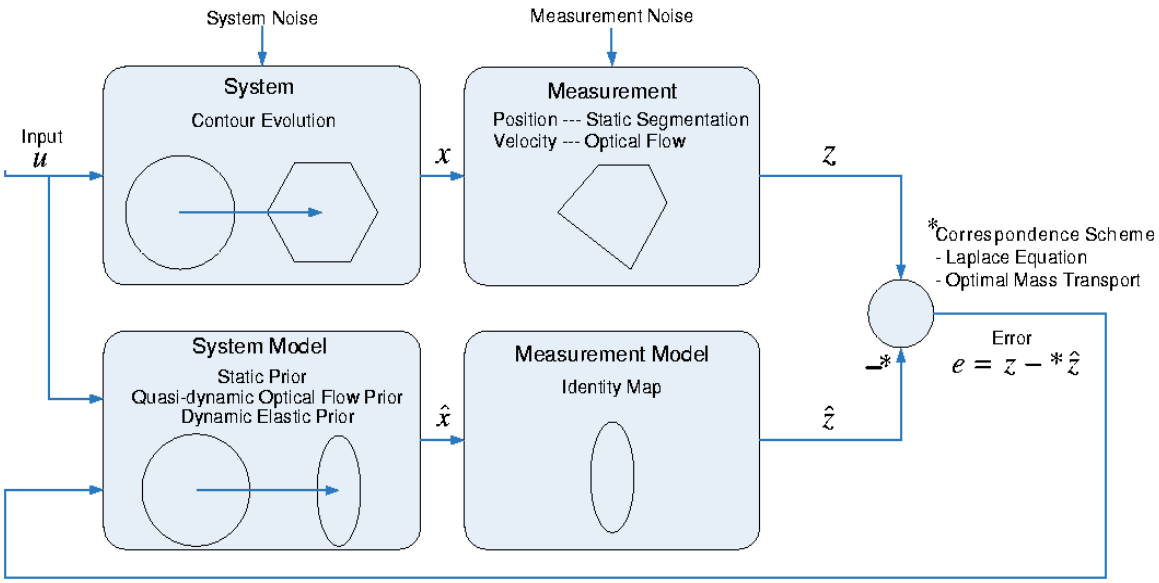


Figure 24: Geometric Observer Structure

the correspondence problem and is applied globally on the implicit representations of the curve. Although in the L^2 case OMT is an NP-hard problem we benefit from a fast and near-realtime, multi-resolution and multigrid implementation of the OMT algorithm on the GPU [81, 80]. Another significant contribution is the use of a registration warp as a velocity measurement instead of the classical Horn & Schunck [55] optical flow used by Niethammer *et al.* [72]. This warp is also a 2D velocity field over every point in the image like its predecessor but has accurate magnitudes making advection unnecessary in the framework and velocity measurements accurate.

The chapter is organized in the following sections. Section 6.2 briefly reviews curve evolution equations and the general observer structure as proposed in [72]. Section 6.3 briefly describes the areas of contribution of this work. In Sections 6.4 to 6.7 we describe our proposed velocity measurement based upon image warping, our proposed error correction scheme based upon optimal mass transport, and the GPU implementation of the framework. Results are presented in Section 6.8. Conclusions are given and future work is discussed in Chapter 7.

6.2 Background

6.2.1 Curve Evolution

A planar curve evolution may be described as the time-dependent mapping: $\mathcal{C}(p,t) : S^1 \times [0,\tau) \mapsto \mathbb{R}^2$, where $p \in [0,1]$ is the curve's parameterization on the unit circle S^1, $\mathcal{C}(p,t) = [x(p,t),\ y(p,t)]^T$, and $\mathcal{C}(0,t) = \mathcal{C}(1,t)$. Define the interior and the exterior of a curve $\mathcal{C}$ on the domain $\Omega \subset \mathbb{R}^2$ as

$$\begin{aligned} int(\mathcal{C}) &:= \left\{ \boldsymbol{x} \in \Omega : (\boldsymbol{x} - \boldsymbol{x}_c)^T \mathcal{N} > 0,\ \forall\, \boldsymbol{x}_c \in \mathcal{C} \right\}, \\ ext(\mathcal{C}) &:= \Omega \setminus \overline{int(\mathcal{C})}, \end{aligned}$$

where $\mathcal{N}$ is the unit inward normal to $\mathcal{C}$. To avoid tracing individual curve particles over time $\mathcal{C}$ can be represented implicitly by a level set function $\Psi : \mathbb{R}^2 \times [0,\tau) \rightarrow \mathbb{R}$ [76], where

$$\Psi(0,t)^{-1} = trace(\mathcal{C}(\cdot,t)).$$

There is no unique level set function Ψ for a given curve $\mathcal{C}$. Frequently, Ψ is chosen to be a signed distance function, defined as follows:

$$\begin{cases} \|\nabla\Psi\| = 1, \text{ almost everywhere}, \\ \Psi(\boldsymbol{x}) = 0,\ \forall \boldsymbol{x} \in \mathcal{C}, \\ \Psi(\boldsymbol{x}) < 0,\ \forall \boldsymbol{x} \in int(\mathcal{C}), \\ \Psi(\boldsymbol{x}) > 0,\ \forall \boldsymbol{x} \in ext(\mathcal{C}). \end{cases}$$

Given a curve evolution equation

$$\mathcal{C}_t = \boldsymbol{v},$$

where $\boldsymbol{v}$ is a velocity vector, the corresponding level set evolution equation is [76]

$$\Psi_t + \boldsymbol{v}^T \nabla\Psi = 0.$$

The unit inward normal, $\mathcal{N}$, and the signed curvature, κ, are given by

$$\mathcal{N} = -\frac{\nabla\Psi}{\|\nabla\Psi\|}, \quad \kappa = \nabla \cdot \frac{\nabla\Psi}{\|\nabla\Psi\|}.$$

6.2.2 General Observer Structure

Assume that the system to be observed evolves in continuous time and that measurements of the system's states become available at discrete time instants $k \in \mathbb{N}_0^+$, i.e.,

$$\begin{aligned}\begin{pmatrix}\mathcal{C}\\ \boldsymbol{q}\end{pmatrix}_t &= \begin{pmatrix}\boldsymbol{v}(\mathcal{C},\boldsymbol{q},t)\\ \boldsymbol{f}(\mathcal{C},\boldsymbol{q},t)\end{pmatrix} + \boldsymbol{w}(t)\\ \boldsymbol{z}_k &= \boldsymbol{h}_k\left(\begin{pmatrix}\mathcal{C}\\ \boldsymbol{q}\end{pmatrix}\right) + \boldsymbol{v}_k = \begin{pmatrix}\mathcal{C}(t_k)\\ \boldsymbol{q}(t_k)\end{pmatrix} + \boldsymbol{s}_k(t),\end{aligned}$$

where $\boldsymbol{w}$ and $\boldsymbol{s}_k$ are the system and measurement noises respectively, $\mathcal{C}$ represents the curve position, and $\boldsymbol{q}$ denotes additional states transported along with $\mathcal{C}$ (e.g., velocities). Assume further a simulation and measurement model of the form

$$\begin{pmatrix}\hat{\mathcal{C}}\\ \hat{\boldsymbol{q}}\end{pmatrix}_t = \begin{pmatrix}\hat{\boldsymbol{v}}(\hat{\mathcal{C}},\hat{\boldsymbol{q}},t)\\ \hat{\boldsymbol{f}}(\hat{\mathcal{C}},\hat{\boldsymbol{q}},t)\end{pmatrix}, \quad \hat{\boldsymbol{z}}_k = \begin{pmatrix}\hat{\mathcal{C}}(t_k)\\ \hat{\boldsymbol{q}}(t_k)\end{pmatrix},$$

where the hat denotes estimated quantities. Assuming $\boldsymbol{h}_k = id$ (the identity map), this corresponds to a completely measurable state $\boldsymbol{x} = [\mathcal{C},\boldsymbol{q}]^T$. The proposed continuous-discrete observer is formally

$$\begin{aligned}\begin{pmatrix}\hat{\mathcal{C}}\\ \hat{\boldsymbol{q}}\end{pmatrix}_t &= \begin{pmatrix}\hat{\boldsymbol{v}}(\hat{\mathcal{C}},\hat{\boldsymbol{q}},t)\\ \hat{\boldsymbol{f}}(\hat{\mathcal{C}},\hat{\boldsymbol{q}},t)\end{pmatrix}, \quad \hat{\boldsymbol{z}}_k = \begin{pmatrix}\hat{\mathcal{C}}(t_k)\\ \hat{\boldsymbol{q}}(t_k)\end{pmatrix}, \qquad (6.2.1)\\ \begin{pmatrix}\hat{\mathcal{C}}_k(+)\\ \hat{\boldsymbol{q}}_k(+)\end{pmatrix} &= \begin{pmatrix}\hat{\mathcal{C}}_k(-)\\ \hat{\boldsymbol{q}}_k(-)\end{pmatrix} +^* \begin{pmatrix}K_k^{\mathcal{C}} *^* \left(\mathcal{C}_k -^* \hat{\mathcal{C}}_k(-)\right)\\ \boldsymbol{K}_k^{\boldsymbol{q}} *^* (\boldsymbol{q}_k -^* \hat{\boldsymbol{q}}_k(-))\end{pmatrix},\end{aligned}$$

where $(-)$ denotes the time just before a discrete measurement, $(+)$ the time just after the measurement, and $K_k^{\mathcal{C}}$ (a scalar) and $\boldsymbol{K}_k^{\boldsymbol{q}}$ (a diagonal matrix of scalars) are the decoupled error correction gains for the curve position $\mathcal{C}$ and the additional state quantities $\boldsymbol{q}$ respectively. The operators $-^*$, $+^*$, $*^*$ denote subtraction, addition, and multiplication of curves and of the additional state quantities propagated with the

curves respectively. They are a crucial part of the proposed observer framework and are implemented implicitly in the error correction scheme described in Section 6.6. Figure 24 shows the overall observer structure as given in Equation (6.2.1) and model assumptions associated with each component of the observer.

6.3 Contributions

We extend the geometric observer framework of Niethammer *et al.* [72] (given above) with image registration techniques such as optimal mass transport in place of the previous methods utilized for curve correspondence and velocity computation. These modifications lead to more accurate tracking performance and overall algorithmic simplification as well as real time implementations on GPU hardware. Specifically, the improvements can be divided into three areas and are described below:

6.3.1 Registration Warp Velocity Measurement

In the original model, classical optical flow (which in general gives incorrect velocity magnitudes) is utilized both as a motion prior and to measure curve velocity. We propose an image warp based on a brightness constancy assumption as a motion prior and curve velocity measurement. This leads to accurate estimation of velocity field magnitudes and orientations, thereby increasing performance and removing the need for advection of state during prediction.

6.3.2 Optimal Mass Transport Error Correction Scheme

In the original model, a non-trivial procedure was needed to carry out filtering of both curve and velocity information involving complex domain decomposition, the solution of the Laplacian, and advections. We solve this problem globally using optimal mass transport as a registration technique between curves. This results in a mapping between curves, removing the need for advection to transport information for error correction. It also alleviates any need for complex domain decomposition and leads

to more accurate tracking performance.

6.3.3 Computational Efficiency.

We leverage the full multigrid PDE solution scheme on the graphics processing unit (GPU; now available on most consumer computers) to solve the PDEs employed in the geometric observer framework resulting in computation times that outperform implementations documented in the recent literature. This shows that the framework is capable of functioning as a real time vision system and that some PDE methods that were previously impractical for real time use are now reasonable to consider in such systems.

In what follows, we describe how the aforementioned contributions are integrated into the observer framework.

6.4 Measurements

6.4.1 Velocity Measurements

Previously in [72], optical flow was used as the velocity measurement $\boldsymbol{v}$ corresponding to the curve velocity state within the geometric observer. In this strategy, a classical Horn & Schunck optical flow [55] is calculated, which estimates the 2D velocities of all image points from frame to frame which are then used as the measurement of curve velocity. However, optical flow is in general inaccurate with respect to flow magnitude and Horn & Schunck optical flow can be inaccurate in terms of flow direction. While both of these facts can lead to poor performance, the former leads to the necessity of advection to transport information for curve prediction; without correct magnitudes information must be transported in a particular direction until it reaches its destination.

As a solution to these problems, we propose the use of a registration warp as a motion prior. Like optical flow, this warp is a velocity field over every point in the

image domain, but has more accurate magnitudes which are suitable for directly mapping one image to another, making advection unnecessary and velocity measurements accurate.

Given two frames of imagery I_1 and I_2, solving for the registration warp $u : \mathbb{R}^2 \mapsto \mathbb{R}^2$ entails minimization of the following energy:

$$\mathcal{L}(I) = \iint \big(I_1(x + \boldsymbol{v}(x)) - I_2(x)\big)^2 dx$$

In other words, we wish to find a u that transforms I_1 so as to match I_2 as closely as possible in terms of squared error.

Such a quadratic minimization can be sufficiently dealt with via the Gauss-Newton method. Leaving derivations aside, the resulting algorithm for computing $\boldsymbol{v}$ is summarized in Algorithm 4 below.

Algorithm 4 Velocity Computation Algorithm

Set $\boldsymbol{v} = 0$.
while Not converged **do**
- Calculate Horn & Schunck optical flow by minimizing the following with respect to $\hat{\boldsymbol{v}}$,

$$\iint \big(I_1(x + \boldsymbol{v}) - I_2(x) + \langle \Delta I_1, \hat{\boldsymbol{v}} \rangle\big)^2 + \lambda ||\Delta \hat{\boldsymbol{v}}||^2 dx,$$

where λ is a constant determining the required smoothness of the resulting field.
- Set $\boldsymbol{v} = \boldsymbol{v} + \rho \hat{\boldsymbol{v}}$, where ρ is the predefined time step for the Gauss-Newton method.

end while

Previously, such minimization would be considered impractical for tracking due to the computational cost involved with step (2). However, this is not the case when computed using a full multigrid scheme on a small, parallel computing architecture such as the graphical processing unit which is now available in most consumer computers (see Section 6.7 for details on the GPU implementation of the warping algorithm).

6.4.2 Curve Position Measurement

Static measurements of curve position provide information about the actual system and allow the model state to converge to the true system states. Motion models used in visual tracking cannot possibly be rich enough to capture the variability seen in deformable objects in natural video sequences. Therefore, it is important that the observer estimate be supplemented with measurements so the observer can adapt to real-life changes in the system.

The estimate model is a function of the curve position and dynamics as it evolves through time. The measurement of the actual system, however, is independent of the model. Therefore, the static curve measurements can be obtained using any segmentation technique with no modification to the observer. Most segmentation methods are either based on local image information from an edge detector, or statistics of regions of image data defined by the curve. Additionally, shape priors can be learned and included in the segmentation algorithm to improve accuracy if detailed shape information is available a priori.

We create our static measurements in a two step process. First, background subtraction is used as a pre-processing step to help separate the target from background clutter. Second, the resulting image is segmented to isolate the target.

In order to obtain a smooth curve C that represents this segmentation, we employ the active contour technique proposed by Chan and Vese [21]. This method attempts minimize the variance over the region inside the curve C, C_{inside} and outside the curve, $C_{outside}$. This is accomplished by minimizing the following energy with respect to C where u and v represent the mean image intensity over C_{inside} and $C_{outside}$ respectively:

$$\int_{C_{inside}} (I-u)^2 dA + \int_{C_{outside}} (I-v)^2 dA + \mu \int C(s) ds$$

This energy is at a minimum when the interior and exterior are modeled best by

u and v. The Chan-Vese segmentation technique is advantageous in that it is robust to noise since image data is integrated over large regions, it's global use of image information makes it robust to different initializations, and its simplicity allows for real time implementation.

6.5 Motion Priors

In [72], three major motion priors were defined: A static motion prior,

$$\hat{\mathcal{C}}_t = \mathbf{0},$$

a quasi-dynamic optical flow prior,

$$\hat{\mathcal{C}}_t = (\boldsymbol{v}_{OF} \cdot \mathcal{N})\mathcal{N},$$

and a dynamic elastic prior,

$$\mu\hat{\mathcal{C}}_{tt} = \left(\frac{1}{2}\mu\|\hat{\mathcal{C}}_t\|^2 + a\right)\kappa\mathcal{N} - (\nabla \cdot \mathcal{N})\mathcal{N} - \frac{1}{2}\mu(\|\hat{\mathcal{C}}_t\|^2)_s\mathcal{T}.$$

For our work, we utilize the latter, as it correctly models the dynamics of the estimated curve $\hat{C}$.

6.6 Error Correction

The observer framework proposed in this paper requires a methodology to associate an estimated curve state to a measured curve state. This amounts to establishing correspondences between points on the measured and the estimated curves. The correspondence map between measured and estimated curves should be diffeomorphic. As we have seen earliers such a one-to-one correspondences between curves can be established using the Optimal Mass Transport technique.

6.6.1 Curve Interpolation

Optimal mass transport can be employed to find correspondences between curves in the following way. Let a curve C be represented as a binary mask as follows:

$$C_{binary}(x) = \begin{cases} 1, & x \in C_{inside} \\ 0, & \text{otherwise} \end{cases} \tag{6.6.1}$$

Given two curves, C^1 and C^2, a correspondence map can be calculated between these curves by finding a mass-preserving mapping (via the algorithm described above) between their implicit representations C^1_{binary} and C^2_{binary}.

The resulting mapping $\tilde{u}^*$ provides a one-to-one correspondence from the implicit represention C^1_{binary} to C^2_{binary}. However, for the purposes of filtering, an intermediate mapping between curves is required. Such an intermediate mapping is easily recovered through the following warping map $u(x, w)$,

$$u^*(x, w) = x + w * (\tilde{u}^*(x) - x), \tag{6.6.2}$$

where $w \in [0, 1]$ the interpolation parameter such that $u^*(x, w = 0)$ is the identity map and $u^*(x, w = 1)$ is the original $\tilde{u}^*$ which maps all the way from C^1_{binary} to C^2_{binary}.

6.6.2 Error Correction

The error correction scheme builds on the results Chapter 2 and subsection 6.6.1. The observer structure dictates that error correction (the combination of measurement and estimate) for the state corresponding to curve position be computed as:

$$\hat{\mathcal{C}}_k(+) = \hat{\mathcal{C}}_k(-) +^* K^{\mathcal{C}}_k *^* \left(\mathcal{C}_k -^* \hat{\mathcal{C}}_k(-)\right),$$

which amounts to curve interpolation via the method from subsection 6.6.1 where $w = K^{\mathcal{C}}_k$.

For the error correction of additional state quantities, state information needs to be exchanged and compared between the measured and the predicted curves i.e.

moved to appropriate positions in the image domain which correspond to the filtered curve $\hat{\mathcal{C}}_k(+)$ to facilitate a pointwise linear combination of values for filtering. This information exchange is performed, again, via the method from subsection 6.6.1 with $w = \boldsymbol{K}_k^{\mathcal{C}}$ to transport velocity measurements $\boldsymbol{q}_i$ & $\hat{\boldsymbol{q}}_i(-)$ to the newly estimated (interpolated) curve $\hat{\mathcal{C}}_k(+)$,

$$\begin{aligned} \boldsymbol{p}_i &= \boldsymbol{q}_i(u^*(x, \boldsymbol{K}_k^{\mathcal{C}})) \\ \hat{\boldsymbol{p}}_i(-) &= \hat{\boldsymbol{q}}_i(-)(u^*_{inv}(x, 1 - \boldsymbol{K}_k^{\mathcal{C}})), \end{aligned}$$

where u^*_{inv} is the inverse map corresponding to u^* and $\boldsymbol{p}_i$ & $\hat{\boldsymbol{p}}_i(-)$ are the velocities on the measured and predicted curves, respectively, transported appropriately to the corrected curve. We can now perform pointwise filtering on the velocity states via,

$$\hat{\boldsymbol{q}}_i(+) = \hat{\boldsymbol{p}}_i(-) +^* \boldsymbol{K}_k^{\boldsymbol{q}} *^* (\boldsymbol{p}_i -^* \hat{\boldsymbol{p}}_i(-)),$$

where $\hat{\boldsymbol{q}}_i(+)$ is the new velocity state estimate.

6.7 GPU Implementation

In this work, the basic building blocks of all the given algorithms (optical flow for image warping, optimal mass transport, and active contour evolution) are the solutions of PDEs. Typically, such problems are computationally intensive and prohibitive for real time tracking implementations. However, in the past few years, it has been shown that graphical processing units (GPUs), which are now standard in most consumer computers and are normally applied to graphics processing for gaming, are particularly suited for several types of parallelizable problems including the solution of PDEs [13, 74]. In our work, we implemented all PDE solvers on the GPU utilizing the full multigrid method, making two algorithms practical for tracking which were not practical before: optical flow for image warping and optimal mass transport. In this section, we give details on the GPU, our implementation, and the computational advantages achieved.

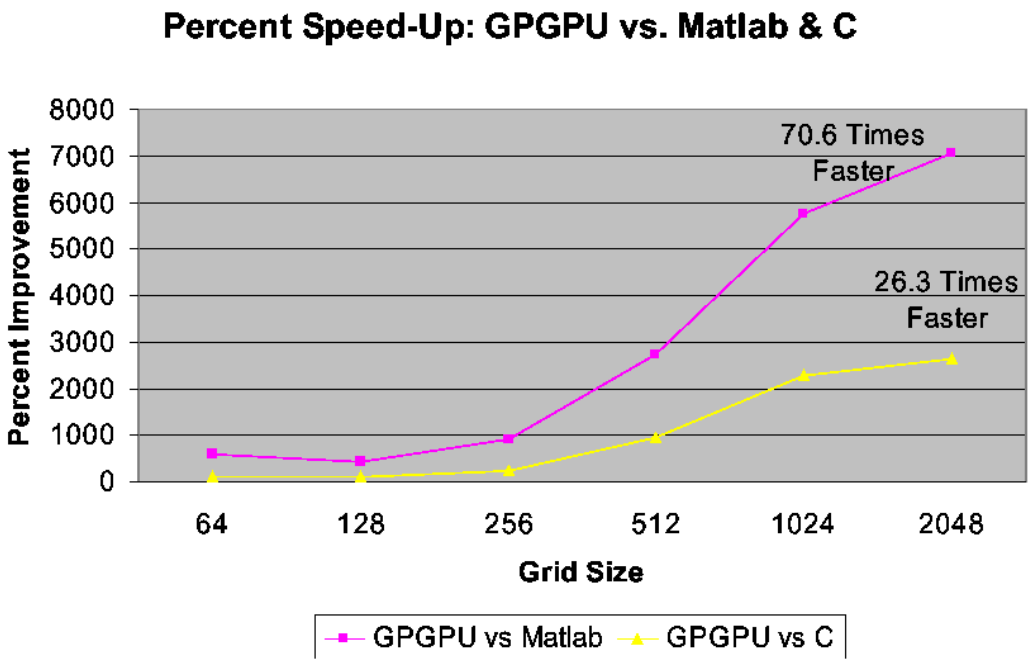


Figure 25: The GPU realizes an increasing advantage in speed over the CPU as grid size increases for the optimal mass transport solver with final performance ratios reaching well beyond an order of magnitude.

GPUs were introduced in Chapter 3 where we provide the details of the method in 3D. In this case, however, both 2D optical flow and optimal mass transport algorithms were implemented on the GPU using a full multigrid solver.

On a modest Dual Xeon 1.6Ghz machine with an nVidia GeForce 8800 GX GPU (3DMark score of 7200) optical flow speeds of 153 FPS were achieved on imagery of size 128^2. This is in contrast to the optical flow framerate of 97 FPS achieved on specialized FPGA hardware recently reported in [29] on comparably sized imagery. Note that the former implementation will run on most modern consumer level PC's without modification.

On the same machine, similar gains in speed were observed for the optimal mass transport algorithm. On a 512^2 grid, 100 iterations of the latest OMT solver required 15.25 seconds (full multigrid in C) while the GPU version required 1.59 seconds representing almost an order of magnitude speed improvement. This gap continues

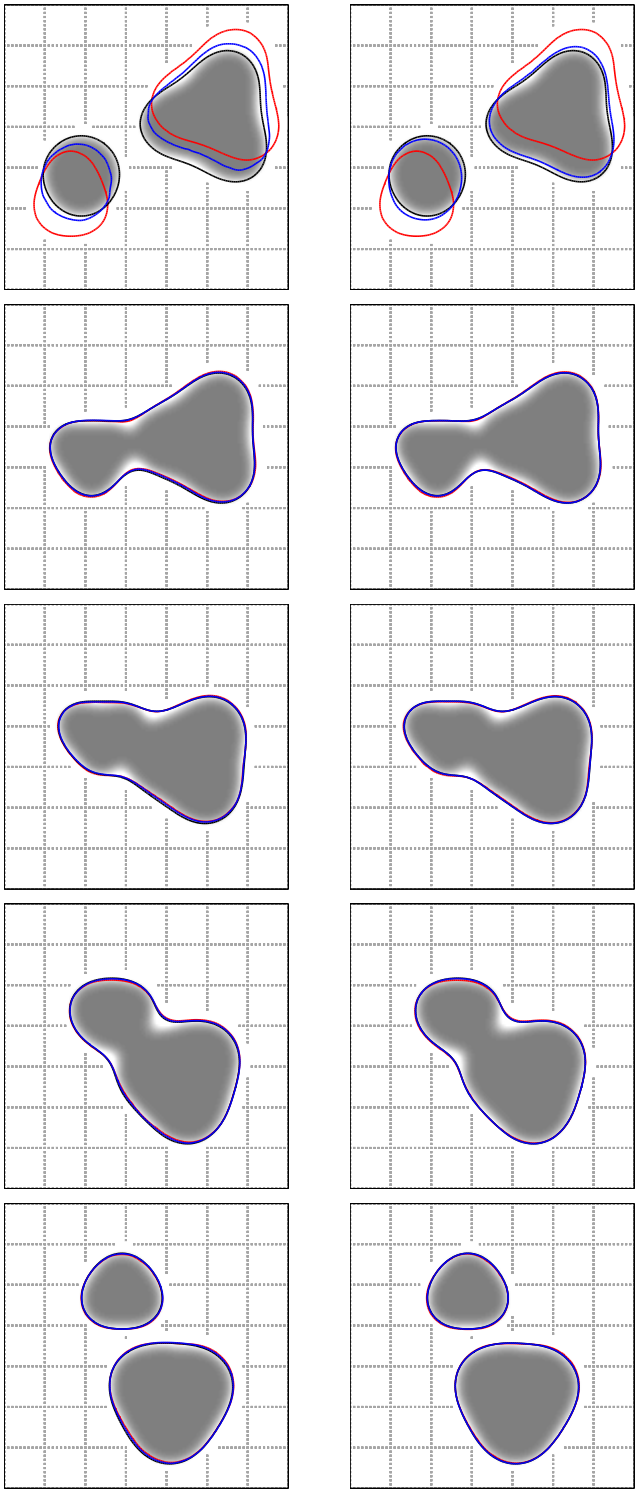
to increase with grid size (Figure 25).

These results show that certain algorithms involving PDEs which would be considered impractical for use in real time control are now reasonable to consider in the design of vision systems. Additionally, these results show that the scheme presented in this paper is capable of real time performance.

6.8 Results

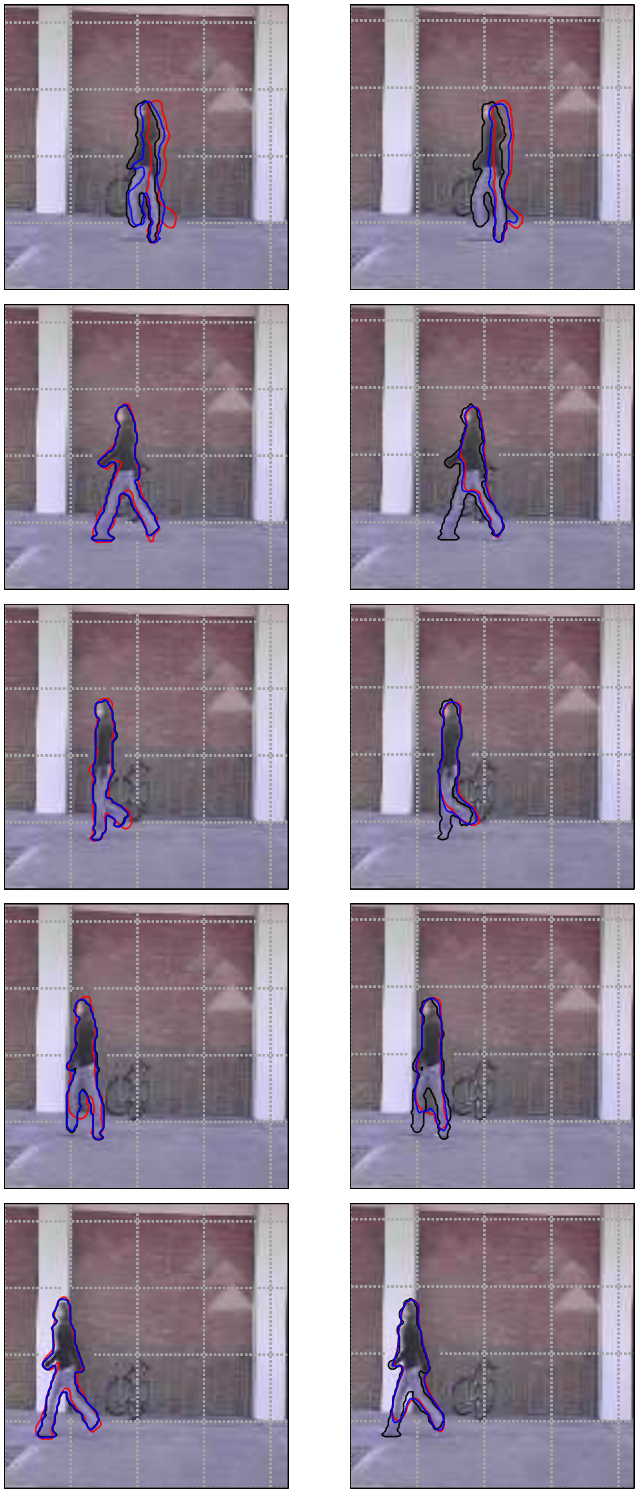
Figures 26 and 27 show tracking results from both the tracker with proposed extensions and the original tracker on a single blob and a walking person, respectively. The error correction gains $K_k^{\mathcal{C}}$ for the curve position and $\boldsymbol{K}_k^{\boldsymbol{q}}$ for the curve's normal velocity were both 0.8 for the blob tracking sequence and 0.3 and 0.5, respectively, for the walking person sequence. Initial conditions (the bold solid curves) were chosen far from the initial measurement curves (dash-dotted curves).

In the blob tracking case, both trackers converge to the correct solution and handle the topological change correctly. This shows that the addition of the optimal mass transport for curve interpolation is capable of handling the same changes in shape the original interpolation scheme could without the need for domain decomposition. In the walking person sequence, the proposed tracker is capable of handling complex shape changes such as those brought on by the moving legs and arms while the original tracker struggles with this difficulty and never recovers.



(a) OMT Interpolation (b) Laplace Interpolation

Figure 26: Comparison of tracking results between the OMT based tracker and the Laplace based tracker on a blob sequence with topological changes. The black curve is the measurement, the red curve is the prediction, and the blue curve is the estimate. Note that both trackers have comparable performance in this simplest of cases showing that our improvements yield no decrease in algorithm capabilities versus the original.



(a) OMT Interpolation (b) Laplace Interpolation

Figure 27: Comparison of tracking results between the OMT based tracker and the Laplace based tracker on a walking person. The black curve is the measurement, the red curve is the prediction, and the blue curve is the estimate. Note that (top) the proposed extensions are capable of handling the complex shape change due to the moving legs and arm while the original algorithm is not able to recover from this initial difficulty.

CHAPTER VII

CONCLUSION AND FUTURE WORK

In this thesis, we presented a novel and efficient numerical method for the computation of the L^2 optimal mass transport mapping in two and three dimensions. Our method uses a direct variational approach. We have formulated a new projection to the constraint technique that can yield a good starting point for the method as well as a second order accurate discretization to the problem. In our future work, we plan to focus on further improvements to our framework by improving the solver for the Karush-Kuhn-Tucker (KKT) system even further. In particular, we plan to develop a multigrid solver for inverting the augmented Lagrangian system on octrees. Moreover, we would investigate migrating the octree data structures to the GPU as well using more modern programming APIs such as CUDA.

In the following we provide our conclusions and planned future work for different applications of our method discussed in this thesis.

7.1 Image Morphing and Registration

Motivated from our results with morphing of natural phenomenon such as flames, clouds and smoke we will further investigate the applications of our algorithm for flame detection and segmentation in image sequences. Moreover, we have also shown that optimal mass transport is, in fact, a viable solution for elastic registration by achieving low run times for typically sized 3D datasets on standard desktop computing platforms. In future work, we will be applying this methodology to other interesting cases as well as extending the results to 3D surfaces (for which the MongeKantorovich theory holds).

We also presented a novel new approach to identify cortical structures by mapping

publicly available 3D MRI atlases to the scan of a patient. OMT is an attractive choice for this because it is a parameter-free algorithm and does not require tuning to specific acquisition sequences. The difficulties of aligning cortical folds is reflected by the large body of literature discussing this topic. Registration approaches based on continuum and fluid mechanics are often applied to this problem. However, the accuracy of these approaches generally depends on how well they are tuned to the sequence of patient scan. We view Optimal Mass Transport (OMT) as part of these types of registration approaches. Unlike the current state of the art, OMT is parameter free. It is, therefore, especially suited to align new acquisition sequences, which the other methods have not yet been tuned to. Based on deriving this numerical framework, we are quite sure that in the near future we will be able to provide cases in which we show superior performance to other well established tools in the community.

7.2 Dynamic Active Contour Tracking

We also presented an extension to the work of Niethammer *et al.* [72] by introducing novel image registration techniques as a velocity measurement and optimal mass transport as a curve interpolation scheme. These additions lead to more accurate tracking performance as well as algorithmic simplifications in the framework. Additionally, we show that the framework and related PDE techniques for image analysis are practical for real time vision applications without the requirement of specialized hardware on consumer level graphical processing units. In fact, we show that the speeds achieved on the GPU exceed those reported in recent literature.

The measurement model applied in this work is a combination of simple background subtraction and Chan-Vese segmentation. Through the utilization of more robust, layered measurement models such as those in our work [78] the applicability of this framework can be expanded now that it is capable of real time performance. Additionally, now that the computational load of the observer framework has been

reduced, it is possible to incorporate a particle filter for non-deterministic estimation.

Additionally, in this work, the interpolation parameter ω was manually set and kept static. An interesting avenue of research would be the investigation of statistical methods for the adaptive tuning of this parameter.

REFERENCES

[1] Ambrosio, L., "Lecture notes on optimal transport problems," *Lectures given at Euro Summer School, July 2000*, 2000.

[2] Angenent, S., Haker, S., and Tannenbaum, A., "Minimizing flows for the monge-kantorovich problem," *SIAM Journal of Mathematical Analysis*, vol. 36, pp. 61–97, 2003.

[3] Arad, N., Dyn, N., Reisfeld, D., and Yeshurun, Y., "Image warping by radial basis functions: Applications to facial expressions," *CVGIP: Graphical Models and Image Processing*, vol. 56, no. 2, pp. 161–172, 1994.

[4] Ascher, U. and Haber, E., "Grid refinement and scaling for distributed parameter estimation problems," *Inverse Problems*, vol. 17, pp. 571–590, 2001.

[5] Ascher, U., Huang, H., and van den Doel, K., "Artificial time integration," *BIT*, vol. 47, pp. 3–25, 2007.

[6] Bangerth, W., *Adaptive finite element methods for the identification of distributed coefficients in partial differential equations*. PhD thesis, University of Heidelberg:Germany, 2002.

[7] Becker, R., *Adaptive Mesh Refinement for Time-Dependent Partial Differential Equations*. PhD thesis, Stanford University, 1982.

[8] Becker, R., *Adaptive finite element methods for optimal control problems*. PhD thesis, University of Heidelberg:Germany, 2001.

[9] Becker, R., Kapp, H., and Rannacher, R., "Adaptive finite element methods for optimal control of partial differential equations," *SIAM Journal of Control and Optimization*, vol. 39, pp. 113–132, 2000.

[10] Beier, T. and Neely, S., "Feature-based image metamorphosis," in *Proceedings of SIGGRAPH*, vol. 26, pp. 35–42, IEEE, 1992.

[11] Benamou, J. D. and Brenier, Y., "A computational fluid mechanics solution to the monge-kantorovich mass transfer problem," *SIAM Journal on Mathematical Analysis*, vol. 35, pp. 61–97, 2003.

[12] Benzi, M., Golub, G. H., and Liesen, J., "Numerical solution of saddle point problems," *Acta Numerica*, vol. 14, pp. 1–137, 2005.

[13] Bolz, J., Farmer, I., Grinspun, E., and Schroeder, P., "Sparse matrix solvers on the GPU: Conjugate gradients and multigrid," in *Proceedings of SIGGRAPH*, vol. 22, pp. 917–924, 2003.

[14] Brandt, A., "Multilevel adaptive solutions to boundary value problems," *Mathematics of Computation*, vol. 31, pp. 333–390, 1977.

[15] Brenier, Y., "Polar factorization and monotone rearrangement of vector-valued functions," *Communications on Pure and Applied Mathematics*, vol. 64, pp. 375–417, 1991.

[16] Briggs, W., Hensen, V., and McCormick, S., *A Multigrid Tutorial*. SIAM, 2000.

[17] Bro-Nielsen, M. and Gramkow, C., "Fast fluid registration of medical images," *Lecture Notes in Computer Science*, vol. 1131, pp. 267–276, 1996.

[18] Bro-Nielsen, M. and Gramkow, C., "Fast fluid registration of medical images," in *Lecture Notes in Computer Science: Visualization in Biomedical Imaging* (Höhne, K. and Kikinis, R., eds.), pp. 267–276, Springer-Verlag, New York, 1996.

[19] Brown, L. G., "A survey of medical image registration," *ACM Computing Surveys*, vol. 24, pp. 325–376, 1992.

[20] Byrd, R. H., Curtis, F., and Nocedal, J., "An inexact SQP method for equality constrained optimization," *SIAM Journal on Optimization*, vol. 19, pp. 351–366, 2008.

[21] Chan, T. F. and Vese, L. A., "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[22] Chartrand, R., Vixie, K., WohlBerg, B., and Bolt, E., "A gradient descent solution to the monge-kantorovich problem," *Submitted to SIAM Journal on Scientific Computing*, 2005.

[23] Christensen, G. E., *Deformable shape models for anatomy*. PhD thesis, Sever Institute of Technology, Washington University, 1994.

[24] Christensen, G. E., Rabbit, R. D., and Miller, M., "Deformable templates using large deformation kinetics," *IEEE Trans. on Image Processing*, vol. 5, pp. 1435–1447, 1996.

[25] Clarenz, U., Droske, M., and Rumpf, M., "Towards fast nonrigid registration. in inverse problems, image analysis and medical imaging," *Lecture Notes in Computer Science*, vol. 313, pp. 67–84, 2002.

[26] Crum, W., Hartkens, T., and Hill, D., "Non-rigid image registration: theory and practice," *British Journal of Radiology*, vol. 77, no. Special Issue, pp. S140–S153, 2004.

[27] Cullen, M. and Purser, R., "An extended lagrangian theory of semi-geostrophic frontogenesis," *Journal on Atmospheric Sciences*, vol. 41, pp. 1477–1497, 1984.

[28] Dean, E. J. and Glowinski, R., "Numerical methods for fully nonlinear elliptic equations of the monge-ampere type," *to appear Computer Methods in Applied Mechanics*, 2008.

[29] Diaz, J., Ross, E., Pelayo, F., Ortigosa, E., and Mota, S., "FPGA-based real time optical flow system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 274–279, 2006.

[30] Eriksson, K., Estep, D., Hansbo, P., and Johnson, C., "Introduction to adaptive methods for differential equations," *Acta Numerica*, pp. 105–158, 1995.

[31] *et al.*, B. F., "Automatically parcellating the human cerebral cortex," *Cerebral Cortex*, vol. 14, no. 11, pp. 11–22, 2004.

[32] *et al.*, R. K., "A digital brain atlas for surgical planning, model-drivensegmentation, and teaching," *IEEE Trans. on Visualization and Computer Graphics*, vol. 2, no. 3, pp. 232–241, 1996.

[33] Evans, L. C., "Partial differential equations and monge-kantorovich mass transfer," *Lecture Notes*, 1989.

[34] Fischer, B. and Modersitzki, J., "Fast inversion of matrices arising in image processing," *Numerical Algorithms*, vol. 22, pp. 1–11, 1999.

[35] Gangbo, W., "An elementary proof of the polar factorization of vector-valued functions," *Arch. Rational Mechanics Anal.*, vol. 128, pp. 381–399, 1994.

[36] Gangbo, W. and McCann, R., "The geometry of optimal transportation," *Acta Mathematica*, vol. 177, pp. 113–161, 1996.

[37] Golub, G. and Greif, C., "On solving block-structured indefinite linear systems," *SIAM Journal of Scientific Computing*, vol. 24, no. 6, pp. 2076–2092, 2003.

[38] Goshtasby, A. A., *2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications*. John Wiley and Sons, Hoboken, NJ, 2005.

[39] Greif, C. and Schotzau, D., "Preconditioners for the discretized time-harmonic maxwell equations in mixed form," *Numerical Linear Algebra Applications*, vol. 14, no. 4, pp. 281–297, 2007.

[40] Gunzburger, M. D., *Perspectives in flow control and optimization*. SIAM, Philadelphia, 2003.

[41] Gupta, M. M. and Zhang, J., "High accuracy multigrid solution of the 3d convection-diffusion equation," *Applied Mathematics and Computation*, vol. 113, pp. 249–274, 2000.

[42] Haber, E. and Ascher, U., "Fast finite volume simulation of 3d electromagnetic problems with highly discontinuous coefficients," *SIAM Journal of Scientific Computing*, vol. 22, pp. 1943–1961, 2001.

[43] Haber, E., Heldmann, S., and Modersitzki, J., "An octree method for parametric image registration," *Journal of Computatinoal Physics*, vol. 223, pp. 783–796, 2007.

[44] Haber, E. and Modersitzki, J., "Multilevel methods for image registration," *SIAM Journal on Scientific Computing*, vol. 27, pp. 1594–1607, 2004.

[45] Haber, E. and Modersitzki, J., "Volume preserving image registration," *Inverse Problems*, vol. 12, pp. 143–152, 2004.

[46] Hajnal, J. V., Hill, D. L. G., and Hawkes, D. J., *Medical Image Registration. The BIOMEDICAL ENGINEERING Series*. CRC Press, Boca Raton, FL, 2001.

[47] Haker, S., Tannenbaum, A., and Kikinis, R., "Mass preserving mappings and image registration," in *MICCAI*, pp. 120–127, 2001.

[48] Haker, S., Zhu, L., Tannenbaum, A., and Angenent, S., "Optimal mass transport for registration and warping," *International Journal of Computer Vision*, vol. 60, no. 3, pp. 225–240, 2004.

[49] Heckemann, R. A., Hajnal, J., Aljabar, P., Rueckert, D., and Hammers, A., "Automatic anatomical brain mri segmentation combining label propagation and decision fusion," *NeuroImage*, vol. 33, no. 1, pp. 115–126, 2006.

[50] Heinkenschloss, M. and Vicente, L., "'analysis of inexact trust region SQP algorithms," *SIAM Journal on Optimization*, vol. 12, pp. 283–302, 2001.

[51] Henn, S. and Witsch, K., "Iterative multigrid regularization techniques for image matching.," *SIAM Journal on Scientific Computing*, pp. 1077–1093, 2001.

[52] Henn, S. and Witsch, K., "Multimodal image registration using a variational approach," *SIAM Journal on Scientific Computing*, vol. 25, pp. 1429–1447, 2003.

[53] Hjaltason, G. R. and Samet, H., "Speeding up construction of quadtrees for spatial indexing," *The VLDB Journal*, vol. 11, pp. 109–137, 2002.

[54] Hoekema, R., Venner, K., Struijk, J., and Holsheimer, J., "Multigrid solution of the potential field in modeling electrical nerve stimulation," *Computers and Biomedical Research*, vol. 31, pp. 348–362, 1998.

[55] Horn, B. K. P. and Schunck, B. G., "Determining optical flow.," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[56] Joppich, W., "A multigrid method for solving the nonlinear diffusion equation on a time dependent domain using rectangular grids in cartesian coordinates," in *Proceedings of the 5th International Conference*, pp. 243–248, 1987.

[57] Kalmoun, E. and Rude, U., "A variational multigrid for computing the optical flow," tech. rep., Friedrich-Alexander University, 2003.

[58] Kantorovich, L. V., "On a problem of Monge," *Uspekhi Matematicheskikh Nauk.*, vol. 3, pp. 225–226, 1948.

[59] Kruger, S. and Calway, A., "Image registration using multiresolution frequency domain correlation," in *British Machine Vision Conference*, pp. 316–325, 1998.

[60] Lee, S.-Y., Chwa, K.-Y., Hahn, J., and Shin, S., "Image morphing using deformation techniques," *Journal on Visualization and Computer Animation*, vol. 7, no. 1, pp. 3–23, 1996.

[61] Lee, S.-Y., Chwa, K.-Y., Hahn, J., Shin, S., and Wolberg, G., "Image morphing using deformable surfaces," in *Proc of Computer Animation*, pp. 31–39, IEEE Computer Society Press, 1994.

[62] Lee, S.-Y., Chwa, K.-Y., Hahn, J., Shin, S., and Wolberg, G., "Image metamorphosis using snakes and free-form deformations," in *Proceedings of SIGGRAPH*, pp. 439–448, 1995.

[63] Losasso, F., Fedkiw, R., and Osher, S., "Spatially adaptive techniques for level set methods and incompressible flow," *Computers and Fluids*, vol. 23, pp. 457–462, 2006.

[64] Losasso, F., Gibou, F., and Fedkiw, R., "Simulating water and smoke with an octree data structure," *SIGGRAPH*, vol. 23, pp. 457–462, 2004.

[65] Maintz, J. A. and Viergever, M. A., "A survey of medical image registration," *Medical Image Analysis*, vol. 2, pp. 1–57, 1998.

[66] McCann, R., "Polar factorization of maps on riemannian manifolds," *Geometric and Functional Analysis*, vol. 11, pp. 589–608, 2001.

[67] Memin, E. and Perez, P., "Dense estimation and object-based segmentation of the optical flow with robust techniques," *IEEE Trans. on Image Processing*, vol. 7, no. 5, pp. 703–719, 1998.

[68] Miller, M., Christensen, G., Amit, Y., and Grenander, U., "Mathematical textbook of deformable neuroanatomies," *Proc. Nat. Acad. of Science*, vol. 90, no. 24, pp. 11944–11948, 1993.

[69] Modersitzki, J., *Numerical Methods for Image Registration.* Oxford University Press, New York, 2004.

[70] Modersitzki, J., Lustig, G., Schmitt, O., and Obeloer, W., "Elastic registration of brain images on large pc-clusters," *Elsevier, Future Generation Computer Systems*, vol. 18, pp. 115–125, 2001.

[71] Moser, J., "On the volume elements on a manifold," *Transactions of the American Mathematical Society*, vol. 120, pp. 286–294, 1965.

[72] Niethammer, M., Vela, P., and Tannenbaum, A., "Geometric observers for dynamically evolving curves," in *Proceedings of the IEEE Conference on Decision and Control, and the European Control Conference*, pp. 6071–6077, 2005.

[73] Nocedal, J. and Wright, S., *Numerical Optimization*. Springer, New York, 1999.

[74] Nolan, G. and et al., "A multigrid solver for boundary value problems using programmable graphics hardware," in *Proceedings of SIGGRAPH*, pp. 102–111, 2003.

[75] Oberman, A., "Wide stencil finite difference schemes for the elliptic monge-ampere equation and functions of the eigenvalues of the hessian," *Discrete and Continuous Dynamical Systems series B (DCDS B)*, vol. 10, pp. 221–238, 2008.

[76] Osher, S. and Fedkiw, R., *Level Set Methods and Dynamic Implicit Surfaces*, vol. 153 of *Applied Mathematical Sciences*. Springer-Verlag, 2003.

[77] Pharr, M., *GPU Gems 2*. Addison-Wesley, 2005.

[78] Pryor, G., Vela, P., Rehman, T., and Tannenbaum, A., "Layered active contours for tracking," in *Proceedings of British Machine Vision Conference*, pp. 22–31, 2007.

[79] Rachev, S. and Rüschendorf, L., *Mass Transportation Problems, Volumes I and II, Probability and Its Applications*. Springer, New York, 1998.

[80] Rehman, T., Pryor, G., and Tannenbaum, A., "Fast multigrid optimal mass transport for image registration and morphing," in *Proceedings of British Machine Vision Conference*, pp. 102–111, 2007.

[81] Rehman, T. and Tannenbaum, A., "Multigrid optimal mass transport for image registration and morphing," in *Proceedings of SPIE Conference on Computational Imaging V*, p. 649810, 2007.

[82] Rohlfing, T. and C. R. Maurer, J., "Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees," *IEEE Trans. on Information Technology in Biomedicine.*, vol. 7, pp. 16–25, 2003.

[83] S, F. M., *Multilevel Adaptive Methods for Partial Differential Equations*. SIAM, 1989.

[84] Saad, Y. and Schultz, M., "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal of Scientific Statistical Computations*, vol. 7, pp. 856–869, 1986.

[85] Sampath, R. S., *A parallel geometric multigrid method for finite elements on octree meshes applied to elastic image registration.* PhD thesis, Georgia Inst. of Tech., 2009.

[86] Silvester, D., Elman, H., Kay, D., and Wathen, A., "Efficient preconditioning of the linearized navier-stokes equations," *Journal of Computational and Applied Mathematics*, vol. 128, pp. 261–279, 2001.

[87] Sundar, H., Sampath, R., and Biros, G., "Bottom-up construction and 2:1 balance refinement of linear octrees in parallel," *SIAM Journal on Scientific Computing*, vol. 30, pp. 2675–2708, 2008.

[88] Szeliski, R. and Lavallee, S., "Matching 3-d anatomical surfaces with non-rigid deformations using octree-splines," *International Journal of Computer Vision*, vol. 18, no. 2, pp. 171–186, 1996.

[89] Szeliski, R. and Lavallee, S., "Motion estimation with quadtree splines," *IEEE Transactions on PAMI*, vol. 18, no. 12, pp. 1199–1210, 1996.

[90] Terzopoulos, D., "Image analysis using multigrid relaxation methods," *IEEE Trans. on PAMI*, vol. 8, no. 2, pp. 129–139, 1986.

[91] Thirion, J.-P., "Fast non-rigid matching of non-rigid images," Tech. Rep. 2547, Project Epidaure, INRIA, France, 1995.

[92] Toga, *Brain Warping.* Academic Press, San Diego, 1999.

[93] Trottenberg, U., Oosterelee, C., and Schüller, A., *Multigrid.* Academic Press, 2001.

[94] Villani, C., *Topics in Optimal Transportation. Graduate Studies in Mathematics. Vol. 58.* AMS, Providence, RI, 2003.

[95] Volkwein, S., "Mesh-independence for an augmented lagrangian-sqp method in hilbert spaces," *SIAM Journal on Control and Optimization*, vol. 38, pp. 767–785, 2000.

[96] Wesseling, P., *An Introduction to Multigrid Methods.* Wiley, 1992.

[97] Wolberg, G., *Digital Image Warping.* IEEE Computer Society Press, 1990.